IMPROVING THE QUALITY OF SERVICE AND SECURITY OF MILITARY NETWORKS WITH A NETWORK TASKING ORDER PROCESS

DISSERTATION

Matthew D. Compton, Captain, USAF

AFIT/DCS/ENG/10-09

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/DCS/ENG/10-09

IMPROVING THE QUALITY OF SERVICE AND SECURITY OF MILITARY NETWORKS WITH A NETWORK TASKING ORDER PROCESS

DISSERTATION

Presented to the Faculty

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

Matthew D. Compton, BA, MA

Captain, USAF

September 2010

AFIT/DCS/ENG/10-09

IMPROVING THE QUALITY OF SERVICE AND SECURITY OF MILITARY NETWORKS WITH A NETWORK TASKING ORDER PROCESS

Matthew D. Compton, BA, MA

Captain, USAF

Approved:

//SIGNED//	
Kenneth M. Hopkinson, PhD (Chairman)	Date
//SIGNED//	
Gilbert L. Peterson, PhD (Member)	Date
//SIGNED//	
James T. Moore, PhD (Member)	Date

Accepted:

//SIGNED//	
M. U. Thomas, PhD	Date
Dean, Graduate School of Engineering and Management	

Abstract

This research presents a Network Tasking Order process that collects mission plans, network capabilities, and historical records to build a Network Tasking Order (NTO). The NTO document directs the form and usage of the network, much like an Air Tasking Order (ATO) directs the usage of air power. The NTO process is fleshed out with the content and format of the NTO given herein for the first time. Tools such as topology control algorithms are then shown through simulation to improve the quality of service of the network by finding favorable ways to connect the assets identified during the NTO process and to route the information through them, in one case preventing a 15% data loss. Furthermore, portions of the network can be hardened against cyber attack through a novel approach to polymorphic networking. The NTO process can provide a complete list of connections that are possible for a network. By periodically changing those connections in use and the routes taken through them, it becomes more difficult for adversaries to map the network in preparation for an attack. In the majority of cases, network availability to an attacker is reduced by more than 50%. It is also shown how existing topology control algorithms can be modified to produce heuristics for polymorphic networking.

Acknowledgments

I first thank my research and academic advisor, Dr. Kenneth M. Hopkinson, for his advice and guidance. I hope my performance as his first Ph.D. student has lived up to his expectations. In addition, I am grateful to my committee members, Dr. Gilbert L. Peterson and Dr. James T. Moore, for all of their input and suggestions.

I also want to thank Michael Dop, Nicholas Gernert, Gabriel Greve, James Haught, Kyle Kavanaugh, and Alexander Stirling for programming assistance and Dr. Dursun Bulutoglu for thoughts on metrics.

Finally, I wish to thank my family. My wife has put up with many late nights, lent a critical eye as proofreader, and provided a never-ending source of love, support, and encouragement. My two sons are my biggest fans, and I'm as proud of them as they are of me. I love you all!

Matthew D. Compton

Production Baby

Nikhil

Table of Contents

List of Figures

Page

List of Tables

Page

List of Symbols

Symbol	Description	Units
a'_{ijf}	An entry in $A'(G)$ – indicates whether (i,j,f) is allowed	N/A
$A'(G)$	The potential-adjacency matrix for network G	N/A
b^k_{ijf}	Additional cost for routing commodity k on (i,j,f)	dollars
c_{ijf}	The fixed cost for including (i,j,f) in E	dollars
cap_{ijf}	The bandwidth capacity of (i,j,f)	Kbps
d^k	The destination node for commodity k	N/A
E	The set of edges chosen for a network	N/A
f	A particular interface type	N/A
F	The number of interface types	interfaces
G	A network, directed graph (N,E), of nodes N and edges E	N/A
$I_{0.5}$	The symmetric Rényi cross entropy	N/A
i	A particular node in N	N/A
(i,j,f)	An edge from node i to node j on interface f	N/A
j	A particular node in N	N/A
k	A particular commodity	N/A
K	The number of commodities for a network	commodities
l	The number of the current polymorphism being solved	N/A
L	The number of polymorphisms being generated	polymorphisms

Symbol	Description	Units
m^k	Binary decision variable for whether commodity k is dropped or not	N/A
n	The number of nodes in N	nodes
N	A set of nodes for a network	N/A
$\boldsymbol{p}$	The second order degree distribution of a network	N/A
$\boldsymbol{q}$	The second order degree distribution of a network	N/A
$\mathbb{R}$	The set of real numbers	N/A
r^k	The bandwidth requirement for commodity k	Kbps
s^k	The source node for commodity k	N/A
t	A time instance	sec, min, hr
t_1	The first of a set of time instances	sec, min, hr
t_2	The second of a set of time instances	sec, min, hr
t_3	The third of a set of time instances	sec, min, hr
u_{if}	The number of interfaces of type f at node i	N/A
v_{ijf}^k	The variable cost for routing commodity k on (i, j, f)	dollars
v_{max}	The maximum value of v_{ijf}^k	dollars
w_i	The degree of node i	N/A
w_j	The degree of node j	N/A
x_{ijf}^k	The percentage of commodity k flowing on (i, j, f)	N/A
$\bar{y}$	The average number of edges for two solutions to a particular MCNDP	N/A
y_{ijf}	Binary decision variable for whether $(i, j, f) \in E$ or not	N/A

Symbol	Description	Units
Δ	A semimetric from $\Omega \times \Omega$ into $\mathbb{R}$	N/A
$\Delta(\omega, \upsilon)$	The semimetric distance between ω and υ	N/A
τ	A generic solution to a particular MCNDP	N/A
υ	A generic solution to a particular MCNDP	N/A
Ω	The semimetric space of all possible solutions to a particular MCNDP	N/A
$\Omega \times \Omega$	The set of all possible pairs of solutions to a particular MCNDP	N/A
ω	A generic solution to a particular MCNDP	N/A
ω_t	A solution to a particular MCNDP at time t	N/A
ω_{t_1}	A solution to a particular MCNDP at time t_1	N/A
ω_{t_2}	A solution to a particular MCNDP at time t_2	N/A
ω_{t_3}	A solution to a particular MCNDP at time t_3	N/A

List of Abbreviations

Abbreviation	Description	1st Use
24 AF	24th Air Force	14
50 NOG	50th Network Operations Group	15
624 OC	624th Operations Center	14
8 AF	8th Air Force	14
ACO	Ant Colony Optimization	30
ACO	Airspace Control Order	20
AF	Air Force	14
AFAPD	Air Force Applications Program Development	51
AF CTO	Air Force Cyber Tasking Order	15
AFI	Air Force Instruction	155
AFIN	Air Force Internet	156
AFIT	Air Force Institute of Technology	202
AFNetOps	Air Force Network Operations	12
AFNOC	Air Force Network Operations Center	203
AFNOSC	Air Force Network Operations and Security Center	154
AFOTTP	Air Force Operational Tactics, Techniques, and Procedures	18
AM	Amplitude Modulation	51
AM	*Ante Meridiem* (before midday)	154
AMD	Air Mobility Division	18

Abbreviation	Description	1st Use
AOC	Air and space Operations Center	96
AOD	Air Operations Directive	19
APAT	Average Percentage Active Time	8
ASN	Abstract Syntax Notation	154
ATM	Asynchronous Transfer Mode	3
ATO	Air Tasking Order	1
AWACS	Airborne Warning And Control System	52
B	Byte	93
BDMLP	Brooke, Drud, and Meeraus Linear Programming solver	72
BestFS	Best-First Search	30
BFS	Breadth-First Search	30
bps	bits per second	101
C2	Command and Control	17
C4 NOTAM	Command, Control, Communications, and Computer systems NOtice To AirMen	14
CCO	Cyber Control Order	15
CI	Confidence Interval	91
COD	Combat Operations Division	18
config.	configuration	121
CONOPT	CONstrained OPTimization	72
CPD	Combat Plans Division	18
CSAR	Combat Search And Rescue	44

Abbreviation	Description	1st Use
CSARTF	CSAR Task Force	114
CST	Central Standard Time	154
DARPA	Defense Advanced Research Projects Agency	33
dB	decibel	51
dBm	decibels relative to 1 milliwatt	51
dcMST	degree-constrained Minimum Spanning Tree	27
DoD	Department of Defense	1
DSN	Defense Switched Network	155
DYNAT	DYnamic Network Address Translation	35
e.g.	*exempli gratia* (for example)	21
EK	Edmonds-Karp	29
ENOSC	Enterprise Network Operations Support Center	154
et al.	*et alia* (and others)	22
eTANG	enterprise Tracking And Notification Graphical user interface	155
etc.	*et cetera* (and so on)	22
ETE	End-To-End	8
FM	Frequency Modulation	51
FSO	Free Space Optical	22
GAMS	General Algebraic Modeling System	72
GIG	Global Information Grid	1
GNTO	GIG NetOps Tasking Order	12

Abbreviation	Description	1st Use
HANC	Hybrid Agent for Network Control	45
H-MANET	Hybrid-MANET	22
HQ	HeadQuarters	85
IA	Information Assurance	33
ICAO	International Civil Aviation Organization	49
IDL	Interface Definition Language	51
IFF	Identification Friend or Foe	44
IP	Internet Protocol	34
ISR	Intelligence, Surveillance, and Reconnaissance	18
JAOC	Joint Air Operations Center	17
JCEOI	Joint Communications Electronics Operating Instruction	42
JETS	JSC Equipment, Tactical, and Space	46
JFACC	Joint Force Air and space Component Commander	17
JFC	Joint Force Commander	17
JIPTL	Joint Integrated Prioritized Target List	20
JP	Joint Publication	204
JSC	Joint Spectrum Center	46
JSF	Joint Strike Fighter	96
JSRC	Joint Search and Rescue Center	96
JSTARS	Joint Surveillance Target Attack Radar System	96
JTF-GNO	Joint Task Force – Global Network Operations	12

Abbreviation	Description	1st Use
JTRS	Joint Tactical Radio System	56
Kbps	Kilobits per second	51
kHz	kiloHertz	51
KL	Kullback-Leibler	37
km	kilometer	97
LaGO	Lagrangrian Global Optimizer	72
LOS	Line-Of-Sight	48
MAAP	Master Air Attack Plan	18
MANET	Mobile Ad-hoc NETwork	2
max.	maximum	121
Mbits	Megabits	89
MCNDP	Multi-commodity Capacitated Network Design Problem	8
MHz	MegaHertz	51
MILCOM	MILitary COMmunications conference	206
MILP	Mixed-Integer Linear Programming	8
min.	minimum	121
min.	minute	86
MITRE	Massachusetts Institute of Technology Research and Engineering	96
MOEA	MultiObjective Evolutionary Algorithm	28
MOS	Measure(s) Of Success	17
ms	millisecond	89

Abbreviation	Description	1st Use
MS	MicroSoft	154
MTO	Maintenance Tasking Orders	15
MTS	Multiple-Time-Scale	51
N/A	Not Applicable	108
NATO	North Atlantic Treaty Organization	15
NAVY_BG	Navy Battle Group	96
NDP	Network Design Problem	22
NetD	Network Defense	15
NetOps	Network Operations	15
NP	Nondeterministic Polynomial [time]	22
NSIRC	National Security Incident Response Center	156
NT	New Technology	154
NTO	NetOps Tasking Order	15
NTO	Network Tasking Order	1
OPNET	OPtimized Network Evaluation Tool	97
OPTASK LINK	OPerational TASKing data LINKs	20
OW	Optical Wireless	31
P2P	Peer to Peer	156
PFP	Pre-Flow Push	29
PNP	Polymorphic Networking Problem	69
QoS	Quality of Service	2

Abbreviation	Description	1st Use
R-H	Rescue Helicopter	96
S1	Source 1	85
S2	Source 2	85
SD	Strategy Division	18
SEAD	Suppression Enemy Air Defense	50
sec.	second	97
SIF	Selective Identification Feature	44
SIPRNET	Secret Internet Protocol Router NETwork	155
SMS	Systems Management Server	155
SPINS	SPecial INStructions	20
StDev	Standard Deviation	95
STO	Space Tasking Order	42
TACAN	TACtical Air Navigaion	44
TACFIRE	TACtical FIRE direction system	51
TACOPDAT	TACtical Operational DATa	20
TBMCS	Theater Battle Management Core System	21
TCNO	Time Compliance Network Order	14
TCP	Transmission Control Protocol	3
TELNET	TELetype (or TELephone) NETwork	35
TET	Targeting Effects Team	18
UAS	Unmanned Aerial System	48

Abbreviation	Description	1st Use
UDP	User Datagram Protocol	34
UHF	Ultra High Frequency	51
US	United States	49
USAF	United States Air Force	203
USMTF	United States Message Text Format	158
USSTRATCOM	United States STRATegic COMmand	12
vs.	versus	33
W	Watt	99
WNW	Wideband Networking Waveform	96
WOC	Wing Operations Center	96

IMPROVING THE QUALITY OF SERVICE AND SECURITY OF MILITARY NETWORKS WITH A NETWORK TASKING ORDER PROCESS

I. Introduction

The Global Information Grid (GIG[1]) as defined by the *Department of Defense* (DoD) *Directive 8100.1* is the "globally interconnected, end-to-end set of information capabilities, associated processes, and personnel for collecting, processing, storing, disseminating, and managing information on demand to warfighters, policy makers, and support personnel" [1:8]. In particular, the GIG includes the military communications network.

This dissertation shows how the employment of a Network Tasking Order (NTO) process can solve some fundamental problems of the GIG. Briefly, these problems include inappropriate topologies formed by wireless networks, a lack of participant awareness of the global structure of the GIG along with the effects of their involvement, and the inability of the GIG to predictively react to participant needs and enemy attacks. These problems are elaborated on shortly.

The NTO is envisioned as an analogue to and offspring of the Air Tasking Order (ATO), the daily tasking of air missions. At the most basic level, the NTO is a document

[1] A list of abbreviations is contained in the prefatory material as an aid to the reader.

that directs the day-to-day operation of specific portions of the GIG, and the NTO process entails the information flow, tools, and personnel required to generate the NTO. The NTO document, in its planning stage (as a pre-NTO), helps expose shortcomings or redundancies in the network. Also, the NTO can be useful in the execution stage to help quickly recover from unexpected events such as finding new routes for traffic when a node fails.

Many subnetworks of the GIG, especially those in a wartime environment, are formed by mobile nodes. Mobile nodes are by necessity wireless, and oftentimes have little infrastructure with which to connect in battlefield milieus. Therefore, nodes must have the capability to connect directly with each other and form what is termed a Mobile Ad-hoc Network (MANET) [2:329]. Some of the advantages of MANETs are that they are rapidly deployable and can be self-organizing. Without infrastructure, each node must act as a router to ensure information can travel between nodes that are not directly linked. Along with the flexibility that MANETs provide, there are also some disadvantages.

Networks involving mobile nodes can have highly dynamic topologies [2:325]. Without careful planning, the topologies that form can suffer from poor quality of service (QoS). Links are short-lived and excessive resources are spent establishing new links [3:3]. The networks may have bottlenecks, or worse, be disconnected [4:150]. Some topologies require data packets to make many hops to travel from a source node to a destination node, which leads to excessive delays. The loss or delay of information can have dire consequences. At the other extreme, a topology may be highly redundant and underutilized. When costs such as battery life, maintenance, or frequency allocation are

considered, this is decidedly wasteful. It is in the DoD's best interest to find a way to mitigate these disadvantages as much as possible. The NTO process is such a way.

In a MANET, individual nodes often make their own decisions on how to connect to the network [2:325]. Unfortunately, what may appear to be the best decision for an individual node may not be in the best interest of the network as a whole. For instance, one particular node may choose to send information through a path that has the fewest hops or the strongest signal, but as a result increases congestion at another node. This potentially results in delays or dropped packets for higher priority data streams trying to route through the same congested node. A congestion control approach such as Asynchronous Transfer Mode (ATM) virtual circuits does not work well with highly mobile military networks, and the strategies employed by Transmission Control Protocol (TCP) make it difficult to guarantee QoS. What is needed is a means to preplan around or to prevent congestion rather than reacting to it. The NTO process is that means.

In *DoD GIG Architectural Vision*, the current GIG is described as static rather than dynamic and incapable of supporting Net-Centric Warfare/Net-Centric Operations [5:1]. The target GIG is depicted as a "unified, agile, end-to-end information enterprise that is protected, optimized, and responsive to user needs." Additionally, "operational GIG capabilities are continually analyzed and provisioned; configurations are controlled; performance is monitored and anticipated; vulnerabilities are mitigated; and resource allocations (including spectrum) are dynamically adjusted to optimize the performance and security of the GIG and meet specific mission demands and priorities" [5:13].

It is important that users can rely upon the infrastructure of the GIG – even while it may be under cyber attack or physical attack. “Redundancy of paths, the ability to reallocate bandwidth based on path conditions, the commander’s policies and priorities, and automated routing alternatives” are listed as keys to the high availability of this infrastructure [5:21]. The NTO process provides those keys.

The GIG is composed of many parts. Individuals and mechanisms charged with network planning need to be, to the greatest extent possible, aware of what composes the GIG, where those pieces are, what their capabilities are, and how they are connected or communicating, both currently and into the future. Possession of this *GIG-awareness* can allow for tactical integration of assets as another planning variable in the battlefield; not unlike logistical considerations such as fuel, ammunition, water, and others used currently in operation planning. There are a variety of ways such awareness can be leveraged to help glean improved performance out of the equipment being used and to aid in protecting that equipment from cyber attack. The NTO process provides this awareness.

The development and application of the NTO process as a means of enhancing *GIG-awareness* entails taking advantage of the highly planned nature of military operations. Military missions require careful planning to ensure appropriate levels of force, synchronization of effort, minimization of risk, and the deconfliction of taskings, airspace, and spectrum. Unlike most MANET research which often relies on random mobility models [6:485; 7:257], in military scenarios the GIG can profit from available

foreknowledge of the general locations of assets, when they will be there, and the type of traffic they will generate.

The QoS of the network is improved by taking advantage of the information collected during the NTO process. Topology control of a MANET involves making the decision of how the nodes in the network are to be connected and what routes the traffic they generate are to follow. The topology control problem is difficult, particularly when directional antennas are used. Nodes must have suitable proximity, correct orientation, and compatible interfaces before they can be linked. One of the basic assumptions for many topology control algorithms is *a priori* knowledge of which links are possible, in other words, who can directly communicate with whom. The information collated during the NTO process can certainly aid in the creation of a matrix indexing this knowledge. After procuring the necessary inputs, topology control algorithms try to minimize the cost of the network while at the same time attempting to get the most high-value traffic through the system.

The security and availability of the network can be improved by dynamically changing the way traffic is routed through the network and perhaps even altering which links are included. This creates what is termed a polymorphic network. The goal is to protect the network from attack by increasing its resistance to mapping by adversaries. The same topology control algorithms used to improve QoS can be modified such that a new topology is in some measurable way different than the previous topology. The algorithms attempt to satisfy this requirement while at the same time trying to optimize

the network. Any spare capacity that is achieved by the topology control algorithms using the pre-NTO as an input can contribute in the traffic shaping.

1.1 Problem Statement

Is it possible to improve the quality of service and security of military networks through the use of a Network Tasking Order process?

1.2 Hypotheses

Owing to the highly planned nature of military operations, it is possible to have advanced knowledge of conditions needed by network planners. By cataloging this information into a pre-NTO, topology control techniques and algorithms can then be employed to optimize the network. In addition, by being aware of what links are possible, routes and connections can be dynamically altered to strengthen the network against cyber attack.

1.3 Research Objectives

The following research objectives are achieved for the first time:

1. The NTO process is developed and described, with examples showcasing content and appearance.
2. Scenarios in which the existence of an NTO process can be shown through simulation to improve the quality of service of a network are provided.
3. A polymorphic networking algorithm is developed and tested.
4. The increased resistance of polymorphic networks to cyber attack is measured.

Not only are the contents of a pre-NTO identified, but an entire NTO process is fleshed out. In particular, the contents and appearance of a finalized NTO document are

demonstrated for the first time with an example tracing the NTO from plan, through production, to execution. Along the way, it is pointed out where topology control can be linked into the process to allow for optimization and improved QoS. It is also shown how existing topology control algorithms can be modified in a unique way to produce a polymorphic network that is resistant to being mapped by adversaries.

1.4 Methodology

The ATO is taken as a primary inspiration and model for the NTO. In addition, the ATO provides much of the data required for NTO generation. The ATO process follows a well-defined series of steps. Arguments are made that the NTO process be performed side-by-side with the ATO process, in the same time frame. Additionally, the format and means of dissemination for the NTO are chosen to match those of the ATO. Along with planning documents such as the ATO, the compiling of a capabilities database and the archiving of historical precedent are shown to be necessary for inputs to the NTO. A list of types of information that would be useful to network planners is given. It is then described how the planners boost QoS by taking the information (collated in a pre-NTO) to perform analysis or to feed as input to topology control algorithms. Another option is to boost security by taking the information as input to polymorphic networking algorithms. The results of the analyses and algorithms are translated into directives which are then published in the NTO.

Once the NTO process is explained and fleshed out, three scenarios are provided to illustrate the potential improvement to QoS that the NTO process can provide. The first scenario shows how the increase in *GIG-awareness* afforded by the NTO process

can prevent a locally made networking decision for a lower priority data source from adversely affecting the flow of a higher priority source. The second scenario uses the foreknowledge of aircraft locations to preplan a route that maximizes throughput and minimizes interference and unnecessary work. The third scenario investigates the decrease in end-to-end (ETE) delay that having an NTO might provide under light and heavy traffic loads for a Combat Search and Rescue mission. In the second and third scenarios, security is improved when messages are directed over specific routes rather than being broadcast to all neighboring nodes.

For developing a polymorphic networking algorithm, Erwin's mixed-integer linear programming (MILP) formulation for solving the multi-commodity capacitated network design problem (MCNDP) is modified. An extra term is placed into the objective function that increases the cost for routing information on an edge over which the information was previously routed. In addition, a new function, Δ, is introduced to measure the difference between two network topologies[2]. For testing the algorithm, 89 different configurations are considered where the number of nodes, the number of interface types, and the number of commodities are varied. For each configuration, up to 10 polymorphisms are generated. The cost of each solution along with its measured Δ distance from the previous solution and the time to solve are tabulated. In addition, the metrics of network diameter and average number of hops are kept.

Finally, to measure the increased resistance of polymorphic networks to cyber attack, the average percentage active time (APAT) is defined. The APAT measures the percentage of time that an attacker listening on a link would expect to find the link active

[2] A list of symbols, such as Δ, is contained in the prefatory material as an aid to the reader.

and not idle. In a static network, any active edges are active 100% of the time and provide an eavesdropper with uninterrupted access to information. The lower the APAT in a dynamically changing network, the less data is likely to be overheard on a random edge. In addition to the APAT, each configuration is tested for edges that are active 100% of the time.

1.5 Results

Once the required inputs and format of the NTO are explained, the NTO process and products are illustrated for the first time. An example is introduced that takes a hypothetical mission from an ATO, pairs it with data from a capabilities database and historical records of similar missions, and collates the information into a pre-NTO for analysis. The example continues with network planners determining a course of action requiring a specific routing of information. The specific route is translated into a network tasking, and the result is shown as it is published in an NTO.

Three scenarios demonstrate the potential improvement to QoS that the NTO process provides. In the first scenario, the high priority source suffered from 3.4518% to 4.7082% loss of data when the lower priority source was allowed to pick its own route without an NTO process. With an NTO process in place, congestion was prevented and neither source lost any data. In the second scenario, without an NTO process, 14.44% of a traffic flow was lost due to interference and 93.75% of the nodes in the scenario received messages intended for a single recipient. With an NTO process, however, 100% of the flow reached the destination with only 31.25% of the nodes involved in the transfer. ETE delays were slightly longer with an NTO-mandated route, but still within

acceptable limits. For the third scenario, ETE delay was found to be about one second longer without an NTO under heavy traffic conditions. Under light traffic conditions, ETE delay was longer with an NTO, but only by about 0.05 seconds.

While developing the polymorphic networking algorithm, two errors in Erwin's MILP formulation and implementation were found and corrected. The consequences of these errors had been detected by several researchers [8, 9, 10], but until now the source was unknown. The new Δ function for measuring the difference between two network topologies is proved to be a semimetric. A large number of test cases were considered for testing the algorithm. A full set of solutions for network configurations up to 20 nodes was generated. A reduced set for configurations of 25-40 nodes was found due to long running times. For 5-20 node configurations, the longest solution time for a single polymorphism was about 10.5 hours. Most polymorphism costs stay within 60% of the cost of the optimal first solution. The Δ function indicates that most of the topologies generated are unique. For 25-40 node configurations, most polymorphisms still stay within 60% of the optimal first solution, with the maximum recorded value of 62.99%.

In terms of the APAT, for networks of 5-20 nodes, there are clear trends. The APAT decreases as the number of interfaces per node increases. As the number of nodes in the network or the number of commodities per node increases, the APAT increases. No configuration with four interfaces per node had any edges that were active 100% of the time. Only 5.28% of configurations with three interfaces per node had edges that were active 100% of the time. For two interfaces per node, 35.56% of configurations had at least one edge that was 100% active. And for one interface per node, 87.78% of

configurations had at least one edge that was 100% active. For 25-40 node configurations, the trends in APAT matched those for 5-20 nodes. Implementing polymorphic networking can clearly improve security, especially when nodes can connect in more than one way.

1.6 Summary

This chapter provides a general introduction to the GIG and some of its fundamental problems. The concept of an NTO is introduced as a solution to those problems. The chapter briefly outlines how the increase in *GIG-awareness* provided by an NTO process may be used to improve the QoS of the GIG through topology control algorithms, and explains a way to extend those algorithms to produce more secure polymorphic networks. The methodology and results of the dissertation are then summarized. Chapter Two (II) introduces the reader to the general areas of Network Tasking Orders, the Air Tasking Order process, topology control, and polymorphic networking while presenting an overview of other research efforts that are related to the problem statement. Chapter Three (III) details the methodology and approach used during this endeavor. Chapter Four (IV) describes in depth the analysis and results that have been achieved. Chapter Five (V) contains the conclusions and recommendations generated from the completed research objectives and analyses.

II. Literature Review

In developing a Network Tasking Order (NTO) process and algorithms for generating a robust network, relevant studies and literature focused on the three main topics of this research are first examined while presenting an overview of other research efforts that are related to the problem statement. First, NTOs are examined as a way of providing the information needed to improve the quality of service (QoS) and security of the Global Information Grid (GIG) and the means of putting into force these improvements. Since the NTO process is closely related to the Air Tasking Order (ATO) process, the ATO's life cycle is then detailed. Next, topology control algorithms are discussed as a tool for optimizing the network to provide the aforementioned improved QoS to the GIG. Finally, the relatively new domain of polymorphic networking is explored with the goal of increased network security in mind.

2.1 Network Tasking Orders

Ranne and McKee advocate that United States Strategic Command's (USSTRATCOM) Joint Task Force – Global Network Operations (JTF-GNO) and/or Air Force Network Operations (AFNetOps) "conduct concept and prototype development with [GIG] Network Operations Tasking Orders (GNTO) as a means for command and control of the GIG" [11:1]. These GNTOs may be used "to communicate not only what to do and who does it with what assets; but also what to monitor and assess" [11:4]. The authors envision three categories of GNTOs [11:5]:

1. Standing Orders: for persistent operational standards,

2. Cyclical Orders: to communicate planning and resource allocation for specific periods of time, similar to an ATO, and

3. Dynamic Orders: through which USSTRATCOM communicates near real-time direction for security and allocation issues.

In a recent email [12], McKee, President of the National Security Cyberspace Institute, indicated that USSTRATCOM leadership liked the idea and is interested in command and control of the cyberspace domain. JTF-GNO has implemented a version of the GNTO that is network defense focused, but not really integrated with anything. It is focused on computer/land networks with "no real thoughts on applicability to air networks or a larger cyber perspective."

The concept of an NTO has also appeared in Stookey [13], which provides "background and data to build a notional battlespace for testing and simulating the use of dynamic networks within the [United States] military" [13:58]. Stookey elaborates on the necessity of developing an NTO to provide dynamic network routers "a basis for making predictive decisions about where given nodes are spatially in a battlespace, what data links might be available, the bandwidth or throughput of such links, the bandwidth requirements of various data flows, and the priority of the data that might be destined to or coming from various nodes" [13:30]. Pecarina [14] pictures an NTO in which a Joint Forces Cyber Component Commander "assign[s] weights of effort to different mission goals in cyberspace" [14:8]. In addition, he sees the NTO as a means of addressing the flip side of the QoS coin mentioned in Chapter One (I). The NTO helps move to a point where information that is not needed or that wastes time, bandwidth, and energy is blocked to allow critical data to get through [14:10].

There are already military documents called NTOs. Due to the present efforts to establish an Air Force (AF) Cyber Command, organizations and responsibilities are being changed on a seemingly regular basis. Prior to August 2009, 8th Air Force (8 AF) AFNetOps had the mission to provide total situational awareness to enable effective command and control of the AF portion of the GIG ensuring air, space, and cyberspace dominance [15]. To accomplish this mission, the commander can issue a Command, Control, Communications, and Computer Systems Notice to Airmen (C4 NOTAM); a Network Tasking Order (NTO); or a Time Compliance Network Order (TCNO) [16:16]. C4 NOTAMs were a process used to disseminate network information to the field and sometimes included TCNOs. The NTO directed "the timely flow of information across the AF-provisioned portion of the GIG. Within the NTO, operational and scheduled events, taskings, and additional information" were presented [16:25]. The AFNetOps commander used the NTO to direct AFNetOps. NTOs were released daily and compliance was mandatory per AF policy. The TCNO was used by the commander of the Air Force Forces - Global Network Operations "to inform responsible AF agencies of network and system vulnerabilities, track implementation of countermeasures, and comply with JTF-GNO taskings" [16:26]. These various documents were more directed towards actions like moving to standard desktop configurations, disallowing thumb drive use, or blocking various file extensions at mail relays. See Appendices A and B for C4 NOTAM and TCNO examples.

The 624th Operations Center (624 OC) was activated in August 2009 along with a new numbered air force, the 24th Air Force (24 AF). The 624 OC's mission is to

"establish, plan, direct, coordinate, assess, command, and control cyber operations and capabilities in support of AF and joint requirements" [17]. The 624 OC is 24 AF's command and control operations center, responsible for the "Air Force provisioned portion of the GIG for the purpose of Network Operations (NetOps) and Network Defense (NetD)" [18]. The 624 OC commander may issue a variety of order types to execute that role. Currently, these order types include a Maintenance Tasking Order (MTO), a NetOps Tasking Order (NTO), and a TCNO. According to Major Matthew Imperial, senior duty officer in the Air Force Network Operations Center, "other orders such as the Cyber Control Order (CCO) and the Air Force Cyber Tasking Order (AF CTO) will likely be used in the future while some of the other orders will be phased out by name and absorbed into one of the remaining orders" [19]. Briefly, MTOs are for cyber maintenance actions of a general nature, NTOs are for critical actions aimed at defending the GIG, and TCNOs are for routine patching of system vulnerabilities. Unofficially, "the AF CTO is an order used to task assigned AF cyber forces to perform specific actions" (such as NetOps, NetD, Network Warfare Support, and Network attack missions) "at specific time frames in support of AF and joint requirements" [17].

Finally, the 50th Network Operations Group (50 NOG) at Shriever Air Force Base publishes a daily NTO through the 22nd Space Operations Squadron to assist in command and control of the AF Satellite Control Network which includes the Defense Support Program, the Navstar Global Positioning System, the Defense Satellite Communications System, NATO III, and Milstar. This schedule makes sure the satellite fliers have the needed ground antenna resources to perform routine tasks and to perform telemetry and

other data transfers [20]. This version of the NTO appears most similar to the product of the NTO process proposed herein, but focuses on a much smaller portion of the GIG.

With several organizations currently using documents called “NTO”, it may be advisable to attach a type designation to the name, if implemented, to avoid confusion. For example, in the future, the name NTO-A may be used to emphasize the relationship the NTO as proposed in this research has with the ATO and to distance itself from the NTO documents produced by the 624 OC and the 50 NOG. Hereafter, the other NTOs are not mentioned, and the term NTO can be taken without confusion to mean the Network Tasking Order as first proposed by Stookey and Pecarina [13; 14].

The various documents described above do not address the core idea of the NTO process – that the network can be optimized and protected through foreknowledge of time, location, activity, and capabilities of the various mobile components of the GIG. While the concept of an NTO may not be new, this research bridges three critical gaps. First, very little detail is given in [11; 13; 14] regarding what the appearance and contents of an NTO should be. Second, there is also little detail concerning how the NTO should be created and disseminated. Finally, the premise that an NTO is beneficial to the QoS and security of the GIG remains to be tested.

2.2 The Air Tasking Order Process

As mentioned in Chapter One (I), the NTO is an analogue to and offspring of the ATO, the daily tasking of air missions. Therefore, it is pertinent to detail the ATO’s life cycle in order to indicate where the NTO fits in. For convenience and clarity, all

definitions are given in terms of a joint environment as opposed to a multinational or pure Air Force environment.

At the highest level, the President and the Secretary of Defense, through the Chairman of the Joint Chiefs of Staff, direct the national effort to ensure the national strategic objectives and joint operation termination criteria are clearly defined, understood, and attainable [21:I-6, I-8]. The strategic objectives, overall end state, and defined measures of success (MOS) are interpreted by the Joint Force Commander (JFC) and translated into a military strategy [22:4-6]. Joint Publication 1-02 defines JFC as "a general term applied to a combatant commander, subunified commander, or joint task force commander authorized to exercise combatant command or operational control over a joint force" [23:285]. The JFC's military strategy consists of objectives, phasing, military end states, and military MOS [22:4-6].

The JFC usually selects a Joint Force Air and Space Component Commander (JFACC) who is in charge of the JFC's overall air interdiction and counterair effort [21:II-17, IV-13]. The operational and tactical control assigned to the JFACC is established by the JFC [24:9]. The JFACC is responsible for "making recommendations on the proper employment of [air forces that are] assigned, attached, and/or made available for tasking; planning and coordinating air operations; or accomplishing such operational missions as may be assigned" [25:73-74]. To that end, the JFACC considers the JFC's military strategy and develops air objectives and phasing, air tasks, measurable end states, and measures of effectiveness for each air task [22:4-6].

The Joint Air and Space Operations Center (JAOC) is "the operational-level command and control (C2) center that provides the [JFACC] with the capability to direct and supervise the activities of assigned and attached forces and to monitor the actions of both enemy and friendly forces" [22:1-1]. The JFACC receives all air and space resources for planning and tasking in the JAOC within the guidance offered by the JFC.

The basic structure of a JAOC consists of five divisions. Air Force Operational Tactics, Techniques, and Procedures (AFOTTP) 2-3.2, *Air and Space Operations Center* describes these five divisions:

> The Strategy Division (SD) concentrates on long-range planning of air, space, and information operations to achieve theater objectives by developing, refining, disseminating, and assessing the JFACC air and space strategy. The Combat Plans Division (CPD) is responsible for near-term air and space operations planning. The Combat Operations Division (COD) is responsible for the execution of the current ATO. The Intelligence, Surveillance, and Reconnaissance (ISR) Division is responsible for providing the JFACC and JAOC with awareness of adversary activity in the battlespace, for coordinating and planning ISR operations that assure awareness of adversary status and activity in the battlespace, and for developing and maintaining targeting information on the adversary. The Air Mobility Division (AMD) plans, coordinates, tasks, and executes the air mobility mission. [22:1-4, 1-5]

The mission of the CPD, in particular, is "to develop detailed plans for air and space operations based on JFC and JFACC-approved guidance received through the SD" [22:4-1]. The CPD's preparation of the ATO provides these detailed plans during the air and space planning and execution process. Completed ATOs are transmitted by the CPD to the COD and units for execution. Typically, the CPD is organized into four core teams with specialized tasks: the Targeting Effects Team (TET), the Master Air Attack Plan (MAAP) Team, the ATO Production Team, and the C2 Planning Team [22:4-1]. The

other JAOC divisions are broken into teams as well, but the focus here is on the CPD. See Figure 1 for the basic structure of a JAOC.

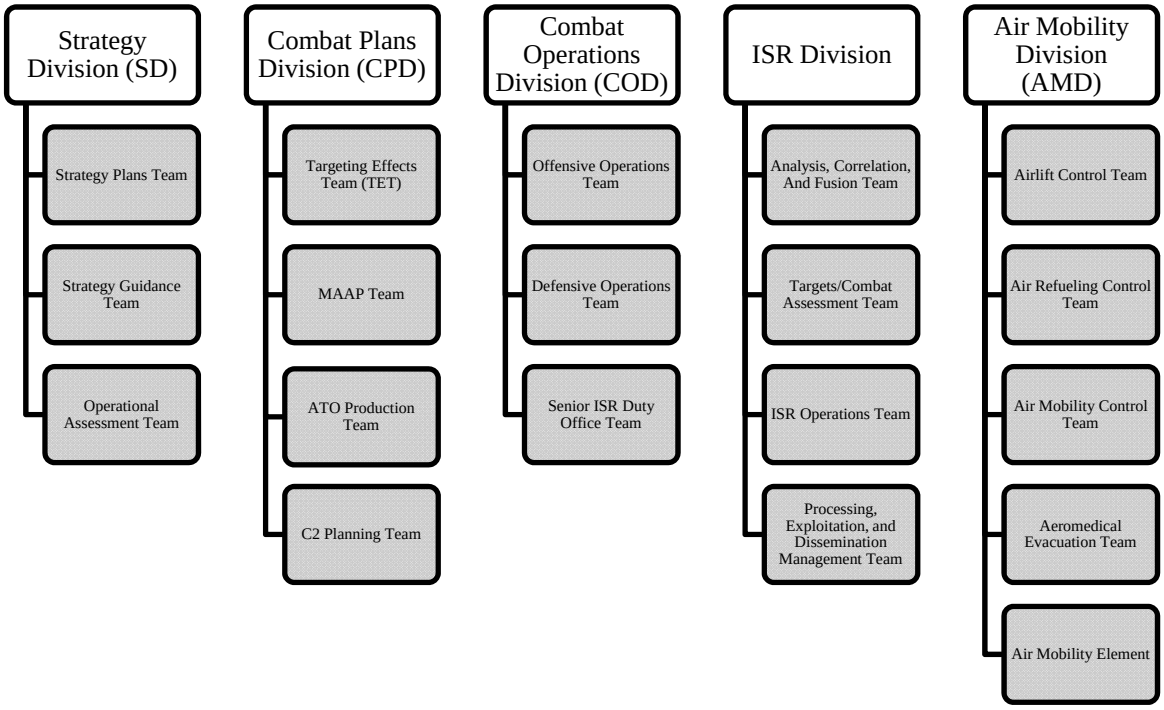

Figure 1: Basic structure of a JAOC [22:1-4]

The daily ATO is the CPD's primary product during execution of air operations. Normally, the CPD works the two ATO periods beyond the current ATO, putting the ATO into a three-day cycle of planning, production, and execution [26:IV-5]. AFOTTP 2-3.2 [22] details the flow of information and intermediate documents involved in creating the ATO. A synopsis of that flow is now given.

To begin, JFACC guidance and apportionment is given to the CPD via the Air Operations Directive (AOD) from the SD. The AOD developed by the SD "combines the JFACC's guidance with the desired targeting priorities to detail [at an operational level]

how the JFACC wishes the air operation to be conducted for the specified [day]. The AOD details what effects are to be undertaken by air operations and the level of effort by force elements, but not how to execute them" [27:5-14].

Based on the guidance in the AOD, the TET drafts a Joint Integrated Prioritized Target List (JIPTL), which must receive JFC/JFACC approval. The purpose of the JIPTL is to align targets with objectives. The JIPTL is sent to the Targets and Combat Assessment Team in the ISR Division to have each apportioned target weaponeered[3]. After weaponeering, the JIPTL goes to the MAAP Team of the CPD to begin the MAAP process.

During the MAAP process, JFACC weapon systems resources are matched to each target. Overall, the CPD is responsible for developing the MAAP, special instructions (SPINS), and the ATO. The MAAP and the ATO are obviously created by the corresponding CPD teams. SPINS is formed by the ATO Production Team with inputs from multiple cells in the JAOC, and is typically included at the end of the ATO. Three other important outputs come from the C2 Planning Team: the daily airspace control order (ACO), the tactical operations data (TACOPDAT), and the daily operational tasking data link (OPTASK LINK) message [22:4-2].

The ACO is used to define and establish special purpose airspace (airspace control means) for management and control. Examples of types of airspace control means are air corridors, air defense areas, reference points, and restricted operations zones. The TACOPDAT is used to "permit the joint operational commander to establish air defense

[3] Weaponeering is "the process of determining the quantity of a specific type of lethal or nonlethal weapons required to achieve a specific level of damage to a given target, considering target vulnerability, weapons characteristics and effects, and delivery parameters" [23:579].

and antiair warfare responsibilities in a tactical area and to permit an area commander to provide supplementary orders for his area of responsibility" [28]. In particular, the TACOPDAT establishes locations and frequencies of ground C2 agencies, combat air patrol stations, airborne early warning stations, airborne radio relay stations, air-to-air refueling stations, and aircraft handover points. The OPTASK LINK specifies data link procedures within a battle group and serves as a list of who can/may talk to whom. It contains information such as unit locations, frequencies, duties, and filter plans [22:4-90].

ISR collection planning and target planning are combined in the MAAP to produce the ATO. After the MAAP process is complete, the ATO data is finally compiled into Theater Battle Management Core System (TBMCS), united with any inputs to SPINS and the ACO, and distributed electronically to all users [22:1-11].

The missions in the ATO can thus be traced back as required to achieve certain effects, which in turn are deemed necessary to meet the JFACC's air objectives. The JFACC's air strategies and objectives are designed to support the JFC's military strategy and objectives, which themselves are intended to support the President's national strategic objectives.

2.3 Topology Control Algorithms

2.3.1 Topology Control Defined

Topology control can refer to different problems depending on the author. Santi defines topology control as "the art of coordinating nodes' decisions regarding their transmitting ranges, in order to generate a network with the desired properties (e.g. connectivity) while reducing node energy consumption and/or increasing network

capacity" [29:30]. Rajaraman defines it as "the problem of computing and maintaining a connected topology among the network nodes" [30:60]. In this dissertation, topology control refers to making a decision on how to design a network as well as determining the routes over which data is to flow. This is closely related to the Network Design Problem (NDP) in which some aspect of the network (cost, diameter, etc.) is to be optimized [31:627-8].

In the context of Mobile Ad-hoc Networks (MANETs), the current set of active links forms the topology of the network. There are many constraints that must be satisfied when designing a wireless topology. Typically, MANETs utilize omni-directional antennas and may communicate with any other node that is within range. However, more realistic network designs for Net-Centric Warfare/Net-Centric Operations include directional wireless technology such as free space optical (FSO) or directed radio frequency devices. A combination of technologies such as these forms a hybrid-MANET (H-MANET). Nodes must have compatible interfaces in order to communicate. Also, orientation of nodes is critical when directional antennas are used. Bandwidth limitations, battery power, and link unreliability due to mobility, weather, interference, or noise must also be taken into consideration. To make matters worse, topology control for networks containing directional links is known to be an NP-hard problem [32:4084; 33:296-297].

2.3.2 Erwin's Mathematical Problem Definition

Erwin [34] used mixed-integer linear programming (MILP) to solve for a topology by expanding on the simpler uncapacitated NDP as formulated by Ahuja, et al. [31:627-8]. Erwin was interested in optimizing the topology of H-MANETs to satisfy the

demands and requirements of military users at a minimum (or near-minimum) cost. His framework can be termed a Multi-commodity Capacitated NDP (MCNDP). The assumptions used for the MCNDP include the following:

1. The number of nodes in the network is fixed.
2. Multiple commodities need to be routed on the network.
3. Each commodity k has a single source node s^k and a single destination node d^k. Note that the use of k in superscript here does not mean exponentiation.
4. Any edge introduced into the network has a fixed capacity that cannot be exceeded.
5. Nodes may have multiple types of interfaces of differing quantities.
6. There may be multiple edges between two nodes provided each edge corresponds to matching interfaces at each node. However, two nodes are limited to one connection per type of interface.
7. Edges are bidirectional with equal capacity in both directions.
8. No edge connects a node to itself.
9. The number of edges incident to a node cannot exceed the total number of interfaces the node has.

The term *commodity* refers to some collection of information that needs to move from a source node to a destination node. In a network with n nodes, there can be up to $n(n-1)$ possible source-destination pairs. In general, there may be multiple commodities between a specified source and destination, each with different service requirements. However, Erwin assumed that commodities are grouped together to determine a unique commodity for each possible source-destination pair.

By allowing multiple interfaces, a network can potentially have many more edges to choose from. Let N denote the set of nodes in the network ($n = |N|$), K the number of

commodities to be routed, and F the number of interface types. It is not enough to denote an edge in the network by an ordered pair of nodes. For the MCNDP formulation, an edge must be specified by an ordered triple (i, j, f) where $i, j \in N$ $(i \neq j)$ and $1 \leq f \leq F$ is the interface type. Let E be the set of edges chosen for the network. The final solution is a network (directed graph) $G = (N, E)$ with routes assigned to each commodity. Since edges are required to be bidirectional, the directed network graphs are typically drawn as undirected graphs with the understanding that each edge actually represents two directed edges.

The setup for MCNDP begins with a potential-adjacency matrix $A'(G) = [a'_{ijf}]$ $(1 \leq i, j \leq n$ and $1 \leq f \leq F)$. Each entry a'_{ijf} is either 0 or 1. A value of 0 indicates that nodes i and j either do not share an interface of type f or that the nodes cannot connect for some other reason (orientation, distance, etc.). A value of 1 indicates that nodes i and j have *potential* to form a link over interface f. Let r^k be the bandwidth requirement for commodity k and let the variable x^k_{ijf} be the percentage of the bandwidth requirement for commodity k chosen to flow on edge (i, j, f). Again, the appearance of k in superscript does not indicate exponentiation, but rather separates the commodity information from the edge information contained in the subscript. This notation conforms to the notation used by Ahuja, et al. [31:627-8]. Each edge (i, j, f) has an associated fixed cost c_{ijf} for inclusion in the network and a usage cost v^k_{ijf} for each commodity for routing 100% of that commodity over that edge. A binary decision variable y_{ijf} indicates whether or not edge (i, j, f) is included in the network design. The number of interfaces of type f available at node i is represented by u_{if}. The capacity of edge (i, j, f) is indicated by

cap_{ijf}. Finally, the expression m^k is a binary decision variable indicating whether or not commodity k must be dropped to achieve a feasible solution. A commodity must be fully satisfied. If it is not possible to route 100% of a commodity's demand, it must be dropped. The formulation for the MCNDP now follows.

Minimize

$$\sum_{(i,j,f)\in E}\left(\sum_{k=1}^{K} v_{ijf}^{k} x_{ijf}^{k}\right)+\sum_{(i,j,f)\in E} c_{ijf} y_{ijf}+\sum_{k=1}^{K} 1000 r^{k} m^{k} \tag{2.1}$$

subject to

$$y_{ijf}=0 \text{ or } 1 \text{ for all } (i,j,f)\in E, \tag{2.2}$$

$$y_{ijf}\le a'_{ijf} \text{ for all } (i,j,f)\in E, \tag{2.3}$$

$$y_{ijf}=y_{jif} \text{ for all } (i,j,f)\in E, \tag{2.4}$$

$$x_{ijf}^{k}\le y_{ijf} \text{ for all } (i,j,f)\in E, 1\le k\le K, \tag{2.5}$$

$$x_{ijf}^{k}\ge 0 \text{ for all } (i,j,f)\in E, 1\le k\le K, \tag{2.6}$$

$$\sum_{k=1}^{K} r^{k} x_{ijf}^{k}\le cap_{ijf} \text{ for all } (i,j,f)\in E, \tag{2.7}$$

$$\sum_{j\in N} y_{ijf}\le u_{if} \text{ for all } i\in N, 1\le f\le F, \tag{2.8}$$

$$m^{k}=0 \text{ or } 1 \text{ for all } 1\le k\le K, \tag{2.9}$$

$$\sum_{\{j,f:(i,j,f)\in E\}} x_{ijf}^{k}-\sum_{\{j,f:(j,i,f)\in E\}} x_{jif}^{k}=\begin{cases}1-m^{k} & \text{if } i=s^{k}\\ m^{k}-1 & \text{if } i=d^{k}\\ 0 & \text{else}\end{cases} \text{ for all } i\in N, 1\le k\le K. \tag{2.10}$$

The objective function (2.1) consists of three terms. The first term adds up the usage costs for routing the various commodities on the network, the second term adds up

the fixed costs for including each edge, and the final term adds a large penalty value for any dropped commodities. The penalty term is included to discourage commodities from being dropped merely to reduce cost. Penalties are assigned in such a way (using r^k) that low bandwidth commodities are dropped preferentially. The coefficient of 1000 was chosen by Erwin [34:28] to be a "very large penalty … so that commodities will be dropped only to achieve feasibility."

Equations (2.2) and (2.9) simply require y_{ijf} and m^k to be binary. Inequality (2.3) only allows an edge to be selected when its entry in the potential-adjacency matrix is 1. Equation (2.4) ensures that links are bidirectional. Inequalities (2.5) and (2.6) force x_{ijf}^k to be between 0 and 1, with positive values allowed only when edge (i, j, f) is used. Inequality (2.7) makes certain that the capacity of each edge is not exceeded. Inequality (2.8) limits the number of edges leaving a node over each interface. Finally, equation (2.10) takes care of conservation of flow requirements with special consideration for dropped commodities.

Inequality (2.3) is actually only implied in Erwin's thesis. This particular constraint was not programmed into his implementation. As a result, subsequent research found unexpected discrepancies between their results and his [8:40; 9:63-64]. The values for x_{ijf}^k, y_{ijf}, and m^k found using this formulation place the network into an optimal arrangement. Optimal solutions are not unique, in general. Ideally, one may find several optimal solutions and rotate among them to create a polymorphic network. Polymorphic networking is described in more detail in the next section.

2.3.3 Solution Methods for the MCNDP

A solution to the MCNDP specifies which edges are to be used, which commodities are to flow, and what routes are to be used to minimize cost. Erwin looked at several approaches to solving this MCNDP formulation. He used a MILP approach using a software program called Xpress-MP, two heuristic strategies, and a combination MILP approach using degree-constrained Minimum Spanning Trees (dcMST).

Using a MILP approach for solving the MCNDP proved problematic. The problem is intractable for deterministic algorithms because, as the number of nodes increases, the problem grows in a non-linear fashion [8:37]. Erwin was unable to obtain a solution using any MILP methods (Newton Barrier, Dual Simplex, Primal Simplex, Branch-and-Bound) in less than 8 hours for networks with 15 nodes or more. Even terminating the MILP search when it finds a feasible solution within 10% of the best known lower bound found in the branch-and-bound process took more than 30 minutes for networks with more than 20 nodes.

The two heuristic strategies employed by Erwin involved first finding a dcMST (using an integer program) to be a network backbone and then adding edges to form a mesh topology. After finding the dcMST, the types and number of unused interfaces are counted at each node. The nodes are sorted based on the number of unused interfaces, then as many links as possible are added while satisfying degree bounds. Heuristic 1 adds links by visiting the nodes in non-decreasing order of the gap between the current degree and the degree upper bound. Heuristic 2 adds links in a similar manner, but visits the nodes in non-increasing order of the gap between the current degree and the degree upper

bound. These heuristics do not consider commodity flows when creating topologies; thus, there must be some post-processing performed to generate the routes. To do so, the MILP formulation of the MCNDP is used omitting the construction costs in the objective function and keeping only the conservation of flow constraint (2.10), the edge capacity constraint (2.7), the constraint that percentage flows must be nonnegative (2.6), and the binary constraint for m^k (2.9). These two heuristics produce inferior topologies in terms of the number of dropped commodities and topology cost. Solutions for networks up to 30 nodes were able to be found in less than 20 minutes, with the majority of the time devoted to the MILP post-processing

Finally, Erwin also looked at a strategy where a dcMST is found first, then the MILP formulation is used with the added constraint that $y_{ijf} = 1$ for all (i, j, f) in the dcMST. This approach allowed for solutions of somewhat larger problems but ends up having the same scaling problems as the pure MILP approach.

Kleeman, et al. [8] approached Erwin's MCNDP formulation using a MultiObjective Evolutionary Algorithm (MOEA) known as the Nondominated Sorting Genetic Algorithm. Since the MCNDP is highly constrained, they realized that standard genetic operators paired with a random initialization process were not likely to generate feasible networks. Thus, care was taken to ensure that chromosomes generated by the operators and the initialization process were able to satisfy the constraints. In particular, a propagation mutation operator was used for each commodity producing a population with 80% of the solutions valid. Search space was also reduced by only allowing the x_{ijf}^k variables to take on values that were multiples of 20% with a heavy bias toward 100%.

They were surprised that they were able to find better solutions to the 10 node instance of the MCNDP than the MILP approach of Erwin. This was surprising because the deterministic MILP approach is supposed to produce an optimal solution. They attributed the probable cause to the optimization software and the lack of parameters specifying the granularity of his commodity flows. Additional problems with Erwin's code have been found; they are discussed in Chapter Three (III). Unfortunately, Kleeman, et al. did not indicate the running time for their approach nor did they provide any information on networks with more than 10 nodes.

Garner [9] developed a set of eight heuristics for finding suboptimal solutions to the MCNDP in reasonable timeframes. His heuristics are based on mapping the MCNDP problem to maximum flow algorithms. Commodities are first sorted and then chosen using one of two methods. The first method uses a dynamic programming solution to the knapsack problem; the second simply uses a greedy approach to pick the next best commodity not already chosen. Every time a new commodity is chosen, it is necessary to calculate whether or not the network can accommodate the flow. This is done using one of two maximum flow algorithms modified for this task. The Edmonds-Karp (EK) algorithm [35:660-3] uses augmenting paths in a residual network. The Pre-Flow Push (PFP) algorithm [36:357-67] increases flow on an edge-by-edge basis. For both maximum flow algorithms, the attempt is made to route commodities using edges that have already been selected. If there is not sufficient capacity, then new edges may be added from potential edges found in $A'(G)$. If a commodity cannot be routed through the network, then it is dropped and the process continues. For both the EK and PFP

algorithms, either a breadth-first search (BFS) or a best-first search (BestFS) is performed for edge selection. The eight heuristic approaches are illustrated in Figure 2 below.

The four heuristics using the dynamic knapsack method were limited by computer memory for storing residual networks and results were only generated for networks with up to 20 nodes. The greedy heuristics can solve networks up to 35 nodes; however, solution quality is not as good and the potential exists for solutions to be trapped at a local minimum. Once networks get up to 35 nodes, the sheer amount of data required as input exceeds the computer's memory capacity. Also, run time for the algorithms continues to be an issue. Greedy commodity selection with EK routing performs the best with run times between 1 and 2 hours for networks with 35 nodes.

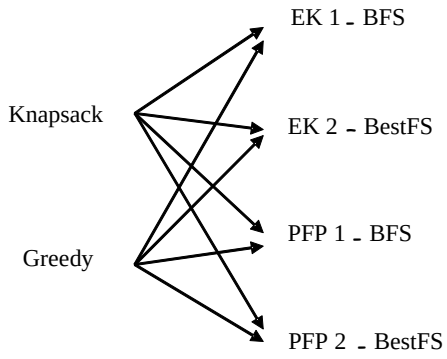

Figure 2: Garner's heuristic approaches for solving the MCNDP [9:46]

The most recent work on the MCNDP was performed by Oimoen [10] on 10-node and 15-node networks. Oimoen utilized ant colony optimization (ACO) algorithms to solve a static version of the MCNDP, a dynamic modification to automatically adjust to a dynamically changing network environment, and a distributed approach to replace a centralized solver. Oimoen's algorithms generated lower average cost solutions than

Garner's four greedy heuristics. The results were comparable with Erwin's 10-node results and better than his 15-node results. Perhaps the most exciting aspect of Oimoen's work is his results with the distributed ACO approach. The distributed algorithms produced comparable results in less than 20% of the computation time.

2.3.4 Similar Topology Control Work

The work of Milner, et al. [37; 38], predates the MCNDP formulation developed by Erwin. They were interested in developing and evaluating low-complexity algorithms and heuristics for Optical Wireless (OW) or FSO sensor networks that look at characteristics such as received power, link fades, signal to noise ratio, and/or network layer delay and choose the best possible topology. The task of reconfiguration requires the formation of a biconnected graph or ring topology. They briefly considered a mixed integer program formulation, but focused on heuristics due to poor scalability. Similar to Erwin's [34] work with the MCNDP, Milner, et al., were interested in proactively adjusting topologies to produce a "better" network, not just attempting to respond to degraded topologies.

The OW networks studied only contain two transceivers per communications node or switch which limits the degree of nodes in the network graph to two. The formulation they use for their congestion minimization problem is very similar to Erwin's MCNDP formulation. Besides the degree two limitation, the main difference between the two formulations is there is no notion of forbidden links and dropped commodities are not allowed. There are four classes of heuristics applied to their formulation: single-hop, multi-hop, rollout, and branch exchange.

For single-hop, the goal is to connect the source and destination of commodities with the heaviest traffic by a single-hop. The effectiveness of this heuristic is marginal due to the degree constraints, and the effect of multi-hop traffic on congestion in the network is not considered. Matching theory can optimally solve this heuristic, but may result in a network that is a disconnected set of rings [38:1559]. The main benefit seems to be a complexity of $O(n^3)$.

In the multi-hop heuristic, commodities are considered in nondecreasing magnitude of required bandwidth. If a single-hop link is available, that link is chosen. Otherwise, the heuristic attempts to create a "least congestion" multi-hop path between the source and destination. This is done by considering all possible choices of links and choosing the one which yields minimum congestion. The complexity for this approach is also $O(n^3)$.

Rollout heuristics are based on the rollout theory in Markov chains. Here, it is used to improvise the ordering of commodities rather than choosing them in the order of bandwidth requirement. Using the ordering determined via rollout, either the single-hop or multi-hop strategies are used to create paths between sources and destinations. This heuristic approach has a complexity of $O(n^5)$.

Finally, the branch exchange is used to consider the result of exchanging two existing links with two new links. Given a topology, a new topology is chosen by picking the topology with the least congestion that results from a branch exchange. The process is repeated until congestion cannot be reduced any more. The complexity for branch

exchange is $O(n^5)$, but is expected to have average runtimes much less than rollout algorithms.

In their evaluation, Milner, et al., found that rollout with a multi-hop strategy for paths produced the best results in terms of congestion. However, multi-hop followed by branch exchange produced much quicker results (6 seconds vs. 69 seconds for a 30-node network).

The concepts explored in this research build mainly upon the work of Erwin and Garner. Major flaws in the software that both Erwin and Garner wrote have been discovered. Erwin's implementation of his MILP formulation was missing a constraint. The code which Garner used suffered from memory leaks, and the data structure he used for keeping track of edges did not allow spare capacity on edges to be used by multiple commodities. The results they generated cannot be trusted, and thus need to be regenerated with corrected software. Details for how the NTO process can link into these topology control algorithms to help optimize the GIG are proposed.

2.4 Polymorphic Networking

2.4.1 DARPA's IA Program and the DYNAT Tool

The Defense Advanced Research Projects Agency (DARPA) initiated the Information Assurance (IA) program in December of 1996 in response to exponential growth of attacks against military networks [39:viii]. The program ended in 2001, with the hope that results established would provide a source of ideas and insight for the community of people working to secure the nation's information systems.

Among other things, the IA program sought research in strategic cyber defense, in particular, using potentially vulnerable components to build intrusion tolerant systems. The overall grand hypothesis of DARPA's IA program was that it is possible to compose trustworthy systems out of less trustworthy components [39:ix]. Five further hypotheses were developed to lead experimentation [40:135]:

1. Layered defenses are an effective means for improving the overall assurance of the system.

2. Dynamic modification of the defensive structure is an effective means for improving the overall assurance of the system.

3. A methodology can be developed which allows useful prediction of risk/system assurance, and constructively supports IA engineering.

4. Automated response mechanisms are at least as effective as human-directed response, qualified by reaction time.

5. Automated decision support functions (e.g., situation awareness, course of action assessment) can provide significant, effective guidance to human operators.

The second hypothesis is of most interest to the research herein. The thought is that "if the network has dynamic characteristics, then the intelligence gathered by the adversary prior to an attack would be time-limited, thus inhibiting the attack" [41:176]. One idea is the use of active network technologies to allow the networks of the future to assist in their own protection. To support the second hypothesis, Kewley, et al. [41], implemented a dynamic network address translation of the Internet Protocol (IP) address and Transmission Control Protocol (TCP)/User Datagram Protocol (UDP) port number combinations in packet headers to defeat network-level analysis tools. Their interest was in network obfuscation. The scheme is to shift the network every so often in order to keep

attackers off balance. Ideally, the shifts cause adversaries to get stuck in their planning phase resulting in them giving up before launching an attack or continuing with high risk and likelihood of failure. Two tests were conducted. The first was designed to demonstrate that it is possible to disrupt an adversary's ability to sniff (capture and analyze) network traffic effectively. The second was to show that improvement of the ability of intrusion detection tools to detect an adversary is viable.

The authors developed a dynamic network address translation (DYNAT) tool that translates addressing information in datagram headers prior to routing to receiver server enclaves. The translation algorithm is cryptographic, having clients and servers configured with initial seed values. A time-based mechanism periodically changes the secret key, synchronized by wall-clock time, and thus changes the translation results. Received datagrams are reverse-translated and forwarded to the correct servers on a private, server-side network interface. No specific encryption algorithm or key exchange technique is required.

For testing, a network topology to model a deployed military campaign-planning unit was built in a lab. Traffic was created both by automated traffic generators and humans at computers. A team from Sandia National Labs played as the adversary with the goals of identifying critical servers through passive sniffing and performing various attacks.

Using this approach, adversary's scans, denial of service attacks, and TELNET hijacking attacks were readily detected and discovery efforts were made substantially more difficult. The DYNAT tool the authors developed had the overall effect of "turning

the tables" on an adversary. Attackers were put into a reactive mode, as opposed to the usual proactive attacking mode. The adversary's ability to identify both the servers and the services they provided was inhibited by the DYNAT approach. Attackers were being effectively thwarted early in their footprinting and scanning steps. These steps are now defined.

McClure, et al. [42], explain various mechanisms of attack and how to protect against them. They discuss the three essential steps that must be performed before a successful hacker can attack: footprinting, scanning, and enumeration. Footprinting involves gathering a profile of a target's Internet, remote access, and intranet/extranet presence. Scanning is the process of identifying live hosts and running services. Finally, enumeration entails probing identified services more fully for known weaknesses. One rather simple step in footprinting is network reconnaissance. Programs such as traceroute[4] can be used to determine network topology and potential access paths into the network. If traffic is encrypted, useful information can still be found in headers. Port numbers, for instance, can help in the scanning step to identify what services are running.

2.4.2 New Polymorphic Networking Approach

This research proposes a novel approach to network obfuscation by dynamically changing a network's topology. Routes in the network can be periodically changed (perhaps in conjunction with the IP/TCP shift) so that traffic patterns change and enemies have a much more difficult time mapping the network. Any topology that is determined by a hacker performing footprinting is short-lived. Additionally, if hackers are

[4] Traceroute allows a user to record the sequence of routers a packet traverses from the source host to a specified destination.

eavesdropping on a particular link, after a short period they may find the link is no longer routing the particular traffic they covet or, better yet, the link may no longer be active.

This approach can be even more effective if there are spare routes or connections going into routers that can be switched among (turning this into a network design problem). However, one must not degrade the network's performance beyond acceptable parameters in the process. The MILP framework for topology control developed by Erwin can be adapted for this purpose. This adaptation requires some means of measuring the difference between two network topologies. More specifically, a function must be given to quantify how different two topologies for the same set of nodes are from each other. Several authors have proposed means for achieving this.

2.4.3 Network Topology Metrics

Krishnamurthy, et al. [43], used a function called the Kullback-Leibler (KL) distance, also known as the relative entropy. It takes as input a stream of data records with two windows maintained over the stream. The windows represent a reference pattern and a test pattern. The KL distance determines whether the two windows differ significantly. This approach examines only one vertex at a time and does not look at the network as a whole. Additionally, the KL distance is not symmetric. That is, the order of comparison can result in different values.

Li, et al. [44], define a normalized metric to differentiate between two graphs having the same vertex set and the same degree distribution. The initial step is to sum, over all the edges of a network, the products of node degrees of the endpoints of each edge. This value is normalized using the maximum and minimum such values for all

simple connected graphs with the same degree distribution. Graphs in which high-degree nodes connect to high-degree nodes and low-degree nodes connect to low-degree nodes tend to have higher values than graphs in which high-degree nodes connect to low-degree nodes. Two graphs with similar normalized metric values are deemed to be similar. Unfortunately, for polymorphic networks, the degree distribution is likely to change and the metric does not apply. Furthermore, the metric does not take into consideration the traffic that flows over the edges. For the purposes of polymorphic networking, two identical graphs with different flow patterns need to be considered different.

Harrington [45] points out that degree distribution is not necessarily a unique representation of a network. He goes on to recommend the use of a second order degree distribution to represent each network in time similar to that in [44]. This distribution is the degree product of the edge-paired vertices. Rather than forming the sum, the symmetric Rényi cross entropy of the second order degree distributions of two graphs is computed. If $\boldsymbol{p}$ and $\boldsymbol{q}$ are the second order degree distributions of a network at two observation points, the symmetric Rényi cross entropy is given by

$$I_{0.5}\left(\boldsymbol{p},\boldsymbol{q}\right) = 2\log_2 \sum_{(i,j)} \sqrt{p_{w_i w_j} q_{w_i w_j}} \tag{2.11}$$

where (i,j) ranges over all edges in the network and w_i and w_j are the degrees of nodes i and j. If $\boldsymbol{p} = \boldsymbol{q}$, then $I_{0.5}(\boldsymbol{p},\boldsymbol{q}) = 0$. The larger the difference between $\boldsymbol{p}$ and $\boldsymbol{q}$, the larger the value of (2.11). This approach avoids the problems in [43]; the measurement is symmetric and looks at the network as a whole. It also does not require a constant degree

distribution, as in [44]. However, the traffic that is flowing over the edges is still not taken into account.

The formula derived in this dissertation is new. It takes both the set of edges being used and the traffic that flows over them into consideration. Additionally, this formula is shown to satisfy all the conditions for being a metric except for the triangle inequality. An original method for generating polymorphic networks using existing topology control algorithms is developed and tested. A special metric is investigated to determine the potential security increase implementing polymorphism may provide.

2.5 Summary

This chapter provided the background and literature review necessary to understand the key concepts to be used in this research. In the first section, a backdrop for the concept of the NTO was set down. The second section explains the life cycle, from strategy to task, of the ATO in the JAOC. In the third section, a discussion on topology control was presented. Erwin's MILP formulation of the MCNDP was given in full detail. Additionally, Erwin's and Garner's heuristic approaches to the MCNDP were explained. The MOEA approach of Kleeman, et al., and the ACO approach of Oimoen were also discussed. The section finished with similar work by Milner, et al., on OW or FSO sensor networks. The fourth section explained the background of polymorphic networking. In particular, the dynamic network address translation approach of Kewley, et al., was spelled out. Also, a few approaches to measuring network change were described. The next chapter details the methodology and approach used during this endeavor.

III. Development and Methodology

The development of a robust Network Tasking Order (NTO) process requires meeting four objectives. The first objective is the development and description of the NTO process. The NTO process uses the Air Tasking Order (ATO) process as a guide. Several illustrations of the steps of the NTO process are crafted to showcase the content and appearance of the NTO and its various inputs. The second objective is to produce scenarios in which the existence of an NTO process can be shown through simulation to improve the quality of service of a network. Three such scenarios are produced. The third objective is to devise a polymorphic networking algorithm that takes its inputs from the NTO process and strengthens a network against cyber attack. To measure the increased resistance of polymorphic networks to cyber attack, the fourth objective is the development of the Average Percentage Active Time (APAT) measure. This chapter has four sections, one for each research objective. In each section, specific information relevant to the objective is specifically discussed. Where appropriate, methodology and design of experiments are detailed. Included are approaches that were taken that did not prove fruitful.

3.1 The Network Tasking Order (NTO) Development Process

The NTO is designed as an analogue to and offspring of the Air Tasking Order (ATO), the daily tasking of air missions. The goal of the NTO process is not to generate network missions to create effects, but rather to support the air missions in the ATO in achieving their designated effects. The NTO process is a means of optimizing the

network comprised of the various assets flying the ATO. This means the NTO process must be done in unison with, the ATO process.

The various planning documents involved in the ATO process are available for use in creating the NTO. In addition to knowing what nodes will be involved in the network, their networking capabilities and expected communication patterns are needed. All of these various inputs are collated into a pre-NTO. Analysis of the pre-NTO is performed to discover any shortcomings in the network, to identify any optimizations that can be carried out, and to strengthen the network against attack. Feedback from the analysis can then be used to make necessary changes to the ATO and other planning documents before they are published. Once sufficient analysis and feedback cycles have been performed, appropriate networking directives are formed and published in a finished NTO for units to download. This NTO process is illustrated in Figure 3.

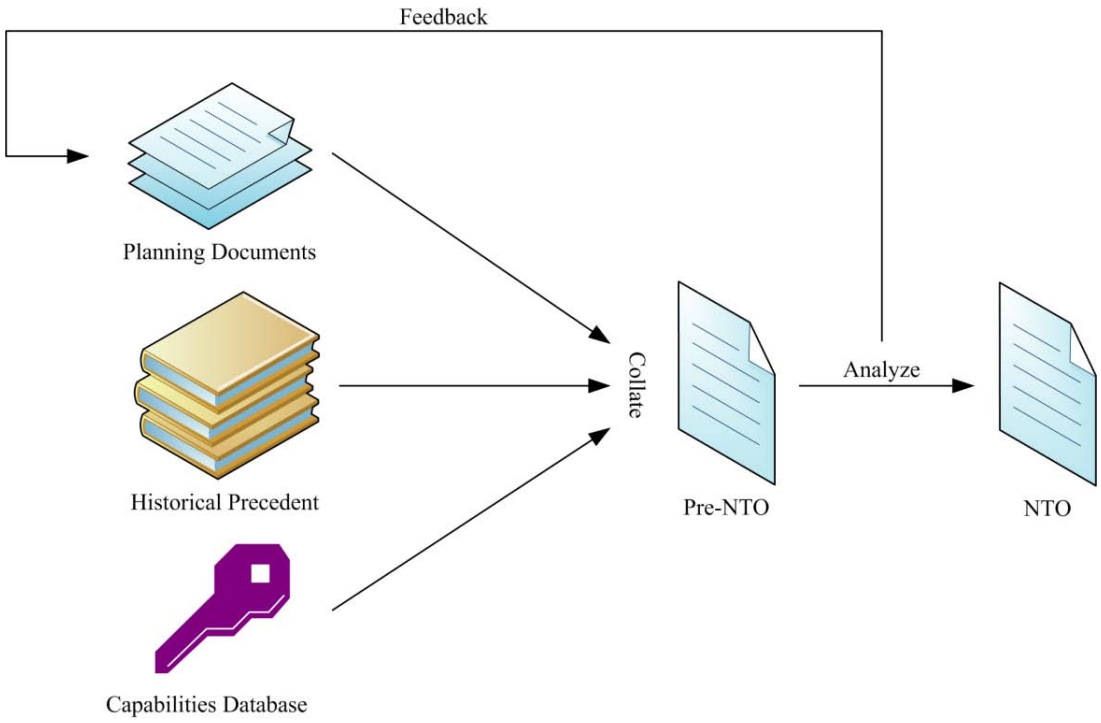

Figure 3: NTO data flow

The three categories of inputs to the pre-NTO are explained in more detail in subsections 3.1.1 through 3.1.3. Subsection 3.1.4 discusses the analysis performed on the pre-NTO. An illustration of the entire NTO process is given in subsection 3.1.5. Finally, other considerations for the NTO process are mentioned in subsection 3.1.6.

3.1.1 Planning Documents

A large portion of the data needed for the NTO comes from the ATO itself and its Master Air Attack Plan (MAAP) predecessor. Specific data from the ATO of relevance to the NTO includes what entities are being tasked, what type of missions they are performing, and the time and location of the missions.

There are many other planning documents such as the Space Tasking Order (STO), Tactical Operations Data (TACOPDAT), Operations Tasking Data Link (OPTASK LINK), and the Joint Communications Electronics Operating Instruction (JCEOI) that contain useful information as well. The STO is another document that is developed in parallel with the ATO, as proposed for the NTO. Its primary purpose is for tasking space assets with specific missions. Since satellites are used for communication, this information is of interest to network designers. The TACOPDAT and OPTASK LINK were described in Chapter Two (II). The JCEOI is used for frequency allocation and deconfliction. The content and structure of the ATO as well as the STO, TACOPDAT, and OPTASK LINK are found in [28], and examples of the message types are provided in Appendices C-F.

To get an idea of the level of detail available to network planners, the structure of an ATO is now broken down with a high-level overview of content. All of the missions

in the ATO are grouped first by tasked country, then by tasked service, and after that by individual tasked units. It makes sense to keep this structure in an NTO so that the units that own each network component can easily find the pieces they are responsible for and configure them for the planned day. In addition, it is possible that communications may need to exist between the asset and its home unit that require planning.

Figure 4 shows a simplified block diagram of how an ATO is organized. For space considerations, only one block on each level is expanded and only three blocks per level are shown. In this figure, there are three tasked countries, the first being the United States. The United States has three tasked services, one of them being the Air Force. The Air Force has three tasked units, among them the 23rd Fighter Squadron. Finally the 23rd Fighter Squadron has been given three missions, one of which has mission number D123HB.

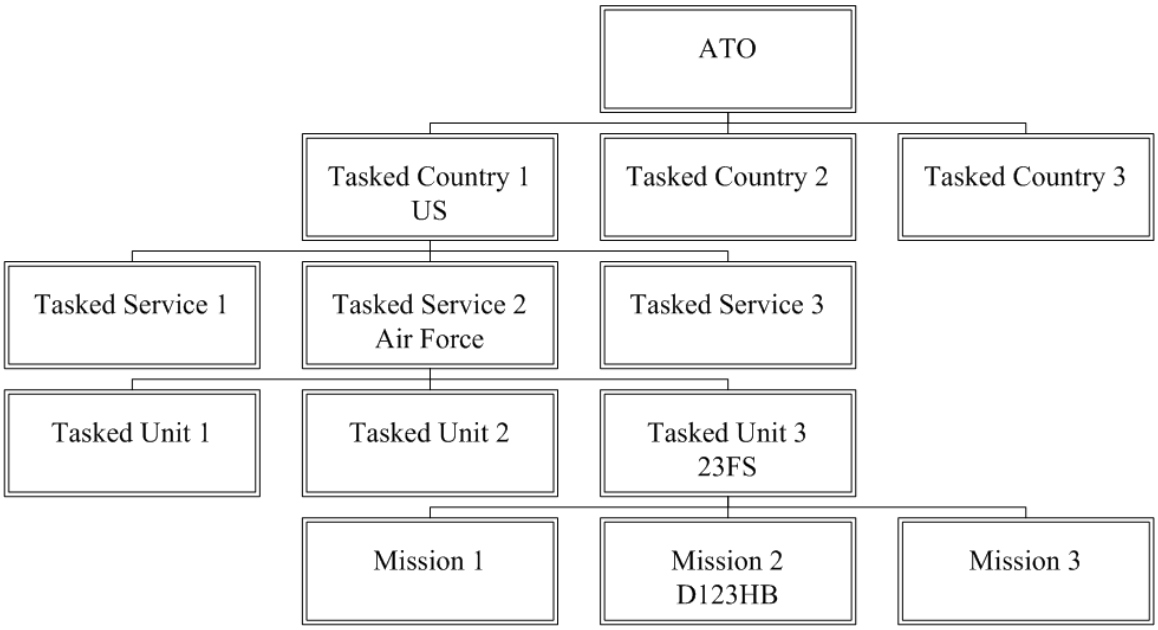

Figure 4: ATO organization

Once the ATO gets down to the individual unit level, all missions that a particular unit is responsible for appear sequentially. Within each mission, the number and type of aircraft along with call sign and primary configuration are given. The following information may also be listed: secondary configuration codes, Link 16 abbreviated call sign, Tactical Air Navigation system (TACAN) channel, primary Joint Tactical Information Distribution System Unit address, and identification friend or foe/selective identification feature (IFF/SIF) mode and code.

Each mission has a preferred mission type or designation. Mission type does not necessarily need to go into the pre-NTO, but it may give a clue as to the types of traffic to expect. For example, a combat search and rescue (CSAR) mission has different traffic characteristics than air reconnaissance or aerial refueling. The expected quantity and burstiness of traffic flows are important measures to include in the pre-NTO. These characteristics can be known through historical precedent. The concept of historical precedent is addressed more fully in subsection 3.1.3 below.

Missions in the ATO usually include a route with altitudes and speeds. Routes can either be a round trip to a target location with departure/return locations and times, one-way travel with departure/arrival locations and times, or orbit information with departure/return locations and times. In any case, given this information, there is some general idea of where an aircraft is going to be and when it will be there. When satellites fly over for limited but predictable time spans, the utilization of these resources can be planned for ahead of time. For example, a directional antenna can be prepositioned to the

expected pointing angle so that it is ready for service when needed. Thus some simplified version of this information is included in the pre-NTO.

Missions in the ATO are also given priorities, and one can generally assume the transmissions of a mission have corresponding priority. Thus, it is imperative to carry these priorities into the pre-NTO to allow for ranking of traffic flows. By including the priority of certain traffic flows, routing agents can use this information to make decisions in situations of congestion. The agents can decide to allow the high priority information to pass through while dropping or delaying the lower priority information. Other alternatives are directing information over different routes, storing information to send at times of lower activity, or requesting nodes to slow down or stop transmissions[5].

As can be seen, there is a great deal of information contained in the ATO that can be used for network planning purposes. The other planning documents such as the STO, TACOPDAT, and OPTASK LINK likewise hold important details. Analysts can know in advance which nodes will be involved in the network and where they will be. This is clearly an advantage over the use of random mobility models. Since the assets forming the nodes are not homogeneous, it is important that network planners know what capabilities those assets have.

3.1.2 The Capabilities Database

There may be differences in networking capabilities between two assets of the same type. For example, the capabilities of an A-10 from one unit may be radically different from those of an A-10 from another unit. However, within a single unit the

[5] Pecarina's Hybrid Agent for Network Control (HANC) is designed for this purpose [14:54-71].

differences are usually minimal. A database of baseline capabilities for each asset type by unit is needed. Some valuable characteristics to include are:

- Types of interfaces (free space optical, radio frequency, etc.),
- Number of interfaces,
- Functional areas implemented for each data link,
- Frequencies or channels available,
- Transmission speed/range,
- Data forwarding capabilities,
- Encryption capability,
- Accepted protocols,
- Set-up time, and
- Queue characteristics.

This capability database needs to be centralized so that when an asset is tasked, the networking capabilities of that asset can be automatically available to the network planners. There does not appear to be an all-inclusive database containing this information currently in existence. Though likely classified, it ought to be fairly straightforward to compile such a database. Once constructed, updates from depots, program offices, or individual units keep the database current. A good basis for building this capabilities database can draw from the Joint Spectrum Center (JSC) Equipment, Tactical, and Space (JETS) database. The JETS database contains detailed technical information about communications, radar, and electronic warfare equipment as well as operational parameters for each subsystem and component [46].

3.1.3 Historical Precedent

The military usually does an outstanding job of performing after-action reviews and cataloging best practices and areas where improvements are needed. The historical qualities of various mission types are certainly tabulated in various locations as lessons learned, listed as general guidelines, or stored in the collective memory of seasoned sergeants. As mentioned above, aspects such as the expected quantity and burstiness of traffic flows for various mission types are important measures to be placed into the pre-NTO. To ensure continuity as personnel change and to increase consistency from NTO to NTO, this historical precedent gets recorded in its own database. As time goes by, the measurements become more refined. After each ATO/NTO is executed, new measurements can be added to the old to extend the usefulness. Periodic reviews ensure consistency and accuracy of the data, identify new metrics to record, and eliminate unused metrics. This also reflects shifts as conditions change over time and from conflict to conflict. Types of metrics to record for each mission type include:

- Expected communications partners,
- Type of data transmitted,
- Bandwidth required (average, burst),
- Quality of service requirements, and
- Encryption needs.

Not only does such documentation help assure that assets receive enough bandwidth for their needs, but it also helps identify where bandwidth is over-allocated. This is especially important as the finite spectrum available gets more and more utilized and

deconfliction becomes more difficult. Additionally, from the network planners' frame of reference, it provides a rough idea of the amount and type of traffic that needs to be routed in the network.

The assets tasked in the planning documents can be cross-referenced with the capabilities database. Historical precedent for the various mission types can be pooled. The combined information can then be collated into a daily schedule for the network. This is a pre-NTO. No network specific taskings have been made yet. Analysis can now be performed upon this pre-NTO.

3.1.4 Analysis of pre-NTO

By having a plan in place, it becomes more apparent where there are single points of failure, gaps in connectivity, or bottlenecks. In such cases, it may be possible to change an orbit or add an extra asset. For example, one waypoint of an E-3's orbit might be adjusted slightly to allow periodic high bandwidth line-of-sight (LOS) communications to a ground unit that is otherwise plagued with constant low bandwidth connections. Another example is the addition of a communications relay mission for an unmanned aerial system (UAS) to linger over a certain location to act as a wireless router in support of a high priority mission.

In addition to eliminating deficiencies, analysis can also be used to optimize communications or to boost security. For example, topology control algorithms can be employed to assign routes that maximize throughput while minimizing the number of links utilized. The topology control algorithms discussed in Chapter Two (II) can be run using the information found in the pre-NTO. Another example might be adding variations

to prevent day to day communication patterns from becoming predictable. The variations may be as simple as rotating frequencies, or more involved like implementing polymorphism.

3.1.4 Example Generation of NTO Tasking

It may be useful at this point to illustrate what a translation of a single mission from an ATO into an NTO might look like. The example provided is academic and does not represent any real mission. In the ATO, each line of data, or set, is terminated by '//'; however, due to length it may wrap to fill multiple lines of text. Within a set, fields are separated by '/'. Fields containing '-' are optional and no data has been entered. Figure 5 shows a few sets that pertain to a single mission from an example ATO.

The first three sets detail whom is being tasked in increasing specificity. The first set indicates that the tasked country (TSKCNTRY) is the US. The second set designates the Air Force (F) as the service being tasked (SVCTASK). The third set specifies the unit being tasked (TASKUNIT) and its location. Here, it refers to the 23rd Fighter Squadron (23FS) at Spangdahlem Air Base in Germany. The location in this instance is given by the International Civil Aviation Organization (ICAO) four character identifier ETAD. Location can also be specified by place name or by latitude/longitude.

```
TSKCNTRY/US//
SVCTASK/F//
TASKUNIT/23FS/ICAO:ETAD//
AMSNDAT/N/D123HB/-/-/-/SEAD/-/-/DEPLOC:KGZ6/241200ZAPR
/ARRLOC:KDZ7/241300ZAPR//
MSNACFT/1/ACTYP:F16CJ/SUPP01/2HARM/-/20001/30111//
AMSNLOC/-/-/-/210/1//
```

Figure 5: Sample ATO mission

The fourth set contains aircraft mission data (AMSNDAT) with twelve fields of information. The first field is the residual mission indicator. An 'N' means that the mission is non-residual[6]. The next field is for the mission number identification, here it is D123HB. The third (Air Mobility Command mission number or event number), fourth (package identification), and fifth (mission commander) fields have been left blank. Field six holds the preferred type or designation for the mission. In this case, it is a Suppression Enemy Air Defense (SEAD) mission. Fields seven (secondary mission type) and eight (alert status) have been left blank. Field nine is used to specify the departure location of the mission if it differs from the location specified in the TASKUNIT set. An ICAO identifier, KGZ6, is given for the location. Field ten gives the day-time and month of departure, April 24 at 1200 Zulu (241200ZAPR). Field eleven provides the recovery location of the mission if other than the location specified in the TASKUNIT set. Again, an ICAO identifier, KDZ7, is given for the location. Finally, field twelve gives the day-time and month of recovery, April 24 at 1300 Zulu (241300ZAPR).

The fifth set holds individual aircraft mission data (MSNACFT) with seven fields of information. The first field gives the number of aircraft as 1. The second field provides the type and model of aircraft as an F16CJ Fighting Falcon. The aircraft call sign, SUPP01, is placed in field three. In the primary configuration code field, 2HARM indicates the aircraft is to be equipped with two AGM-88 high-speed anti-radiation missiles. Field five (secondary configuration code) has been left blank. Fields six and seven are both for IFF/SIF codes. In field six, 20001 indicates a mode 2 code (personal

[6] A non-residual mission falls entirely within a single ATO period.

unit identity) with octal value 0001. In field seven, `30111` indicates a mode 3 code (normal air traffic control identity) with octal value 0111.

Finally, in the sixth set, additional mission location (`AMSNLOC`) information is given with five fields. This set provides mission location information for missions which have no specific target location, for example orbits or alerts. Fields one (day-time and month of start), two (day-time and month of stop), and three (mission location name) are left blank. Field four provides the vertical distance in hundreds of feet above mean sea level. A value of `210` indicates the mission is to fly at 21,000 feet. The last field is the code for the priority assigned to a mission, which in this example is `1`.

Notice that in a few short lines, there is some pretty detailed information about where this particular F-16CJ will be and when it will be there. Given the departure and recovery locations of the mission, intermediate locations can be interpolated. Similar information is given for all other aircraft flying during the same time period. Suppositions can also be made regarding with whom this aircraft will be communicating.

Suppose that a capabilities database contains the following (hypothetical) information regarding F-16CJs from the 23rd Fighter Squadron:

- Equipped with one Improved Data Modem (IDM-302) capable of 16 Kbps digital communications over four independent channels accepting AFAPD, TACFIRE, IDL, and MTS protocols [47]

- The IDM-302 is interfaced with an AN/ARC-164 Ultra High Frequency (UHF) Airborne Radio which receives AM signals at levels between -101 dBm and +2 dBm and features 1-10 watts AM, 100 watts FM, 25 kHz channel spacing over a frequency range of 225.000 to 399.975 MHz, and LOS voice [48]

- The aircraft includes an omni-directional UH-408 UHF Blade Antenna with -1 dB gain over 225-400 MHz and 50 Ohm impedance [49].

Finally, based on historical records of SEAD missions, an average or expected data rate along with a peak data rate can be established. For the sake of the example, suppose the average traffic rate is 5 Kbps with bursts up to 15 Kbps. It is also possible to predict with whom the F-16CJ will be communicating. There will likely be communications between various aircraft in the same strike package. These other aircraft are identified in the ATO. Also, there will be communications with an E-3 Sentry Airborne Warning and Control System (AWACS) providing situational awareness of the battlefield and battle management. The AWACS closest to the F-16CJ during this mission can be identified from the ATO. There may also be communications between the F-16CJ and its home base and with the Joint Air and Space Operations Center (JAOC).

All of this information is of interest to analysts who are trying to optimize the network and eliminate potential gaps in connectivity or bottlenecks. This information all becomes collated into a pre-NTO. The data in the pre-NTO can then be fed into software tools such as the topology control algorithms discussed in this dissertation, or other programs that may exist or have yet to be created. The pre-NTO may have the appearance of Figure 6.

```
TSKCNTRY/US//
SVCTASK/F//
TASKUNIT/23FS/ICAO:ETAD//
AMSNDAT/N/D123HB/-/-/-/SEAD/-/-/DEPLOC:KGZ6/241200ZAPR
/ARRLOC:KDZ7/241300ZAPR//
MSNACFT/1/ACTYP:F16CJ/SUPP01/2HARM/-/20001/30111//
AMSNLOC/-/-/-/210/1//
MODEMDAT/1/TYPE:IDM302//
RADIODAT/1/TYPE:ARC164//
IFACEDAT/1/TYPE:UH408//
EXPCOMM/AVE:5/BURST:15/1//
COMMLINK/ICAO:ETAD/ICAO:JAOC/CALL:SKYWATCH43//
```

Figure 6: Sample pre-NTO mission

Notice that the first six sets in Figure 6 are repeated from the ATO. This is important information, most of which is of use to analysts. Five new sets have been added using the same format as the ATO. The first added set, MODEMDAT, provides information on the modem(s) carried by the asset. The first field in this set gives the number and the second field gives the type. Here it lists 1 IDM-302 modem. These fields can be repeated if there are multiple types of modems. The second and third added sets have a very similar format. The RADIODAT set gives the number and type of radio(s) on the asset. The IFACEDAT set lists the number and type of interfaces (or antennas) being used. It may be necessary to make these sets hierarchical if, for instance, different modems are paired with different radios and antennas. These three sets are derived from the information from the capabilities database.

The last two added sets are derived from historical precedent. The EXPCOMM set lists information on the expected communications traffic from this asset on this particular mission. The first field gives the anticipated average traffic rate in Kbps. The second field gives the anticipated maximum traffic rate in Kbps. The final field indicates the priority of the communications. In this case, the priority is identical to the mission priority listed in the AMSNLOC set of the ATO. This set is also a good place to put fields such as encryption requirements, type of traffic (voice, video, telemetry, etc.) being sent, protocols being used, and so on. The COMMLINK set lists some of the main communications partners for the asset. The field in this set is repeated for each expected traffic recipient. The labels in the field, ICAO and CALL, specify how the recipient is identified. The first two fields in this example give the ICAO identifiers for the F-16CJ's home

station and the JAOC (ICAO:ETAD and ICAO:JAOC). The last field gives the aircraft call sign of an AWACS that is to be coordinating the mission (CALL:SKYWATCH43).

Finally, an example can now be shown of a tasking that is generated and given to this particular mission. Suppose that it has been determined that during this mission, the F-16CJ must route all traffic destined for the JAOC through a KC-135 tanker with call sign FUEL03 in an orbit named BLUE 23 in the vicinity of the mission. Figure 7 shows how this tasking may appear.

```
TSKCNTRY/US//
SVCTASK/F//
TASKUNIT/23FS/ICAO:ETAD//
TASKNODE/ACTYP:F16CJ/SUPP01//
1RTEDAT
/DEST         /START       /STOP        /NHOP          /LOC
/ICAO:JAOC    /241200ZAPR /241300ZAPR /CALL:FUEL03 /BLUE 23//
```

Figure 7: Sample NTO tasking

Notice that the first three sets in Figure 7 are repeated from the ATO and the pre-NTO. These sets narrow down what is being given the tasking. The fourth set, TASKNODE, specifies which node in the network is being tasked. Here it is the F16CJ with call sign SUPP01. The last three lines form what is known as a columnar set. Columnar sets are arranged in vertical columns under an appropriate column heading. The first line has the set name, which for columnar sets must begin with a number. Here the set is named 1RTEDAT, indicating that the set is used for specifying data routes. The second line has the column headers which designate the type of information located in each column. The third line contains the information for this set. The information in this line is entered so that it falls under the proper column headers. The first column, DEST, indicates the final destination for the traffic that is being routed. The final destination

here is referenced by the ICAO code for the JAOC (`ICAO:JAOC`). The next two columns, `START` and `STOP`, specify the day-time and month span over which this route is to be used. Here, the time span corresponds to the full mission duration found in the ATO, from 1200 to 1300 Zulu on 24 April. The fourth column, `NHOP`, indicates the next hop for traffic. The KC-135 is referenced using its call sign (`CALL:FUEL03`). The final column, `LOC`, indicates the location of the next hop. Here, `BLUE 23` refers to the KC-135's orbit. Additional columns can be added as needed. For example, channel, encryption type, interface, etc. can be included in this set. As many information lines as necessary can be used with columnar sets. This set resembles a routing table.

3.1.5 Other NTO Process Considerations

The Air Force uses Theater Battle Management Core System (TBMCS) to assist in planning and executing ATOs. TBMCS contains two integrated databases that can be accessed by the individual squadrons that are being tasked [50]. Each unit may filter an ATO to display only the information that is pertinent to its own missions; however, the entire ATO is available and can be saved as a text file. As an ATO day progresses, certain missions may need to be changed or cancelled and new missions created. These changes are published through TBMCS and are available to a unit until its missions are flying. Once a mission is in the air, any changes to that mission are simply communicated as needed over radio or through text messages. It is entirely possible for an aircraft to be flying a significantly different route than was last published in the ATO. If NTOs are created in conjunction with the ATO, then it makes sense to employ TBMCS to disseminate the NTO and its changes as well.

Once a mission is underway, the Joint Tactical Radio System (JTRS) works in conjunction with tools such as FalconView and Link 16 to allow pilots to see other entities on their missions in a common operating picture. JTRS is a family of software-programmable tactical radios that provide the warfighter with voice, data, and video communications [51]. FalconView is a mapping system that displays various types of maps and geographically referenced overlays. In particular, the various waypoints and orbits from an ATO can be displayed along with the user's own flight plan and the current positions of other entities in the vicinity. Link 16 is one of the nine waveforms that are implemented in JTRS. Link 16 allows for LOS exchange of tactical pictures in near real time.

Using the tools mentioned above, deviations from the ATO and NTO can be visualized. If the location of an aircraft reported through Link 16 does not match with the expected trajectory displayed in FalconView, then a deviation is occurring. Suppose, for instance, that the KC-135 tanker referred to in the NTO tasking of Figure 7 is moved to a different orbit. If that change is made prior to 1200 Zulu, then it is conceivable that a change to the NTO can be published before the F-16CJ leaves on its SEAD mission. Either a new next hop is provided, or the orbit location is updated in the 1RTEDAT set. On the other hand, if the F-16CJ begins its mission and does not find the KC-135 where it was expected, there may be problems. If it is still in range, the transmissions to the JAOC can still be routed through the tanker. If it is not in range, some new route needs to be established. Existing route discovery protocols can find this new route, or a request for

route assistance can be broadcasted. Researching these various scenarios is outside the scope of this dissertation, but makes excellent future research topics.

3.2 Scenarios Utilizing the NTO Process

Three scenarios have been devised to illustrate the potential improvement to quality of service (QoS) that following the NTO process provides. The first shows how the increase in *GIG-awareness* afforded by the NTO process can prevent a locally made networking decision for a lower priority data source from adversely affecting the flow of a higher priority surveillance source [52]. The second scenario uses the foreknowledge of aircraft locations to preplan a route that maximizes throughput and minimizes interference and unnecessary work [53]. The third scenario investigates the decrease in end-to-end (ETE) delay that having an NTO might provide under light and heavy traffic loads for a Combat Search and Rescue (CSAR) mission [54; 55]. These three scenarios are detailed in Chapter Four (IV).

3.3 Devising and Testing a Polymorphic Networking Algorithm

Erwin's mixed-integer linear programming (MILP) formulation for solving the multi-commodity capacitated network design problem (MCNDP) has been successfully adapted to this purpose. Erwin used an optimization program called Xpress-MP to solve the MILP problem using the Newton Barrier method, the Primal Simplex method, and the Dual Simplex method. Xpress-MP is part of a suite of mathematical modeling and optimization tools used to solve linear, integer, quadratic, non-linear, and stochastic programming problems [56]. There are two adaptations needed to solve this problem:

1. Make the process periodic, so that after a determinate amount of time, a new solution is generated.

2. Introduce a mechanism to encourage new solutions to be measurably different than the previous solution.

Maintaining the trappings of Erwin's formulation assures that the aims of minimizing costs while maximizing performance are upheld. An added benefit is that Erwin's formulation has already been implemented and tested, and multiple heuristics for solving the problem more quickly have already been developed. The first adaptation is relatively simple to implement. The entirety of Erwin's formulation can be placed into a loop with some means of controlling the frequency at which solutions are generated. At the end of each loop, the current solution needs to be stored so that the next iteration can be compared to it. If necessary, a timer can be employed to generate new solutions on a regular (or irregular) basis. For comparison, the DYNAT tool remapping rates used by Kewley, et al., were on two-minute and five-minute intervals [41:181].

The second adaptation involves defining a means of measuring the change between two topologies to establish bounds on how different each new solution is from the previous result. Care must be taken that changing the topology does not increase the cost or decrease the performance by more than some acceptable amount. The changes must not be made haphazardly. Initially, topology control algorithms can place a network into an optimal arrangement. Optimal solutions are not unique, in general. Ideally, one makes a change to a new topology that is also optimal. Unfortunately, even if multiple optimums exist, the difference between two optimums may be too small to be of consequence or so large that switching between the solutions throws the network into

disarray. The better approach is to find a way to control the extent of the change of the network.

3.3.1 The Δ Semimetric

The MILP framework for topology control developed by Erwin for the MCNDP is the basis for the dynamically changing topology control algorithms being evolved. Therefore, the formula for measuring topological difference is defined using his symbology. Let Ω denote the set of all possible solutions to a particular MCNDP. For a specific $\omega \in \Omega$, ω consists of the set of values chosen for the variables x_{ijf}^{k}, y_{ijf}, and m^{k}. Since solutions are to be generated periodically over time, ω_t is used to denote a solution found at time t. To avoid a surplus of subscripts, the values of the variables for ω_t are represented as functions of time. The development of a formula to quantify the difference between two solutions ω_{t_1} and ω_{t_2} is now given.

Subtracting $x_{ijf}^{k}(t_1)$ from $x_{ijf}^{k}(t_2)$ measures the difference in the percentage of required bandwith for commodity k that flows on edge (i, j, f) from solution ω_{t_1} to solution ω_{t_2}. Because an increase in flow on one edge must correspond to a decrease on another edge, some of these differences are positive and others are negative. To avoid cancellation of terms, the absolute values of the differences are taken. These absolute differences are summed over all possible edges in the network to obtain a measure of how much the route for a single commodity has changed. Note that two dissimilar changes can result in the same measured difference. Also, the measure is zero if and only if the route is unchanged.

Once these measures are computed for each commodity, the weighted average is taken using the bandwidth of each commodity as the weight. This enforces the idea that a change made to a large bandwidth commodity is more significant than a similar change made to a small bandwidth commodity. This notion is important because the values for x_{ijf}^{k} are percentages. For instance, changing the flow of a 100 Kbps commodity from 50% to 60% over an edge is more consequential than changing the flow of a 10 Kbps commodity from 50% to 60% on the same edge. Finally, the weighted average is divided by the average number of edges in the network at times t_1 and t_2. In this way, a change in a small network is more significant than the same change in a large network. Changing two edges in a network with 20 edges is more consequential than changing two edges in a network with 200 edges.

Using the notation as described above, define

$$\Delta\left(\omega_{t_1},\omega_{t_2}\right)=\frac{\sum_{k=1}^{K}\left[r^{k}\cdot\sum_{i,j\in N}\sum_{f=1}^{F}\left|x_{ijf}^{k}\left(t_2\right)-x_{ijf}^{k}\left(t_1\right)\right|\right]}{\left[\sum_{k=1}^{K}r^{k}\right]\cdot\left[\sum_{i,j\in N}\sum_{f=1}^{F}\frac{y_{ijf}\left(t_2\right)+y_{ijf}\left(t_1\right)}{2}\right]} \tag{3.1}$$

to be the difference between two solutions to the MCNDP formulation found at times t_1 and t_2. It is now shown that (3.1) satisfies three of the four requirements to qualify as a metric or distance function under the assumption that values for r^k remain constant. This is a reasonable assumption, for it ensures that two solutions refer to the same MCNDP formulation and hence belong to the same set Ω.

To qualify as a metric, Δ must be a function from $\Omega \times \Omega$ into $\mathbb{R}$ that satisfies the following conditions:

1. $\Delta(\omega, \upsilon) \geq 0$ for all $\omega, \upsilon \in \Omega$.
2. $\Delta(\omega, \upsilon) = 0$ if and only if $\omega = \upsilon$.
3. $\Delta(\omega, \upsilon) = \Delta(\upsilon, \omega)$ for all $\omega, \upsilon \in \Omega$.
4. $\Delta(\omega, \tau) \leq \Delta(\omega, \upsilon) + \Delta(\upsilon, \tau)$ for all $\omega, \upsilon, \tau \in \Omega$.

The first condition is called the positivity condition; the second, nondegeneracy; the third, the symmetry condition; the fourth, the triangle inequality [57:47]. If Δ satisfies all conditions except the triangle inequality, Δ is said to be a semimetric, and Ω is a semimetric space (*halbmetrischen Raum*) [58:115].

Clearly, Δ is a function from $\Omega \times \Omega$ into $\mathbb{R}$ by definition. Since none of the variables or constants in the formula for Δ are negative, and the only subtraction occurs within absolute value bars, positivity is satisfied. For nondegeneracy, both directions of the implication must be shown. Obviously, if $\omega_{t_1} = \omega_{t_2}$, then $\Delta(\omega_{t_1}, \omega_{t_2}) = 0$, because all of the absolute differences are 0. Contrapositively, assume $\omega_{t_1} \neq \omega_{t_2}$. Then one or more of the following must be true: $m^k(t_1) \neq m^k(t_2)$ for some $1 \leq k \leq K$; $y_{ijf}(t_1) \neq y_{ijf}(t_2)$ for some $i, j \in N$ and $1 \leq f \leq F$; or $x^k_{ijf}(t_1) \neq x^k_{ijf}(t_2)$ for some $i, j \in N$, $1 \leq f \leq F$, and $1 \leq k \leq K$. The third case plainly results in $\Delta(\omega_{t_1}, \omega_{t_2}) \neq 0$ because at least one of the absolute differences is nonzero. However, both the first and the second case imply the third case. For if $m^k(t_1) \neq m^k(t_2)$ for some $1 \leq k \leq K$, then commodity k is dropped in one solution but not the other. Thus, x^k_{ijf} is zero for all edges in one solution and nonzero for at least one edge in the other solution. Likewise, if $y_{ijf}(t_1) \neq y_{ijf}(t_2)$ for some $i, j \in N$ and $1 \leq f \leq F$, then edge (i, j, f) is used in one solution but not the other. Thus, some commodity k flows on edge (i, j, f) in one solution but not the

other. Therefore, $x_{ijf}^{k}(t_1) \neq x_{ijf}^{k}(t_2)$ for this k. Symmetry easily follows from the fact that $|a-b| = |b-a|$ and $a+b = b+a$ for all $a, b \in \mathbb{R}$ (absolute value and addition are symmetric).

The triangle inequality is the one missing property that prevents Δ from being a proper metric unless additional assumptions are introduced. A simple counterexample, as illustrated in Figure 8, shows that the triangle inequality does not hold for Δ in general. Consider a three node network with nodes labeled 1, 2, and 3. There is a single commodity from node 1 to node 2 with a bandwidth requirement of 1 Kbps. Let $\omega \in \Omega$ be a topology that sends 100% of this commodity directly from node 1 to node 2. Let $\upsilon \in \Omega$ be a topology that sends 50% of the commodity directly from node 1 to node 2 and 50% of the commodity from node 1 to node 3 to node 2. Finally, let $\tau \in \Omega$ be a topology that sends 100% of the commodity from node 1 to node 3 then to node 2. Formula (3.1) results in $\Delta(\omega, \upsilon) = 0.375$, $\Delta(\upsilon, \tau) = 0.3$, and $\Delta(\omega, \tau) = 1$. However, $1 \nleq 0.375 + 0.3$.

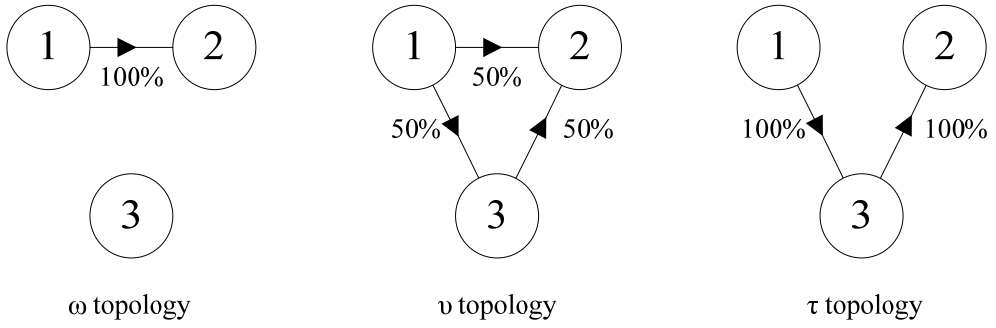

Figure 8: Triangle inequality counterexample for Δ

Under certain conditions, it is possible for the triangle inequality to hold. For instance, if the average number of edges is assumed to remain constant over time at some

value $\bar{y}$, then the argument as shown in Figure 9 can be made. An explanation of the steps is given in the next paragraph.

$$
\begin{aligned}
\Delta\left(\omega_{t_1},\omega_{t_2}\right)+\Delta\left(\omega_{t_2},\omega_{t_3}\right) &= \frac{\sum_{k=1}^{K}\left[r^k\cdot\sum_{i,j\in N}\sum_{f=1}^{F}\left|x_{ijf}^k\left(t_2\right)-x_{ijf}^k\left(t_1\right)\right|\right]}{\left[\sum_{k=1}^{K}r^k\right]\cdot\left[\sum_{i,j\in N}\sum_{f=1}^{F}\frac{y_{ijf}\left(t_2\right)+y_{ijf}\left(t_1\right)}{2}\right]}+\frac{\sum_{k=1}^{K}\left[r^k\cdot\sum_{i,j\in N}\sum_{f=1}^{F}\left|x_{ijf}^k\left(t_3\right)-x_{ijf}^k\left(t_2\right)\right|\right]}{\left[\sum_{k=1}^{K}r^k\right]\cdot\left[\sum_{i,j\in N}\sum_{f=1}^{F}\frac{y_{ijf}\left(t_3\right)+y_{ijf}\left(t_2\right)}{2}\right]}\\
&= \frac{\sum_{k=1}^{K}\left[r^k\cdot\sum_{i,j\in N}\sum_{f=1}^{F}\left|x_{ijf}^k\left(t_2\right)-x_{ijf}^k\left(t_1\right)\right|\right]}{\left[\sum_{k=1}^{K}r^k\right]\cdot\bar{y}}+\frac{\sum_{k=1}^{K}\left[r^k\cdot\sum_{i,j\in N}\sum_{f=1}^{F}\left|x_{ijf}^k\left(t_3\right)-x_{ijf}^k\left(t_2\right)\right|\right]}{\left[\sum_{k=1}^{K}r^k\right]\cdot\bar{y}}\\
&= \frac{\sum_{k=1}^{K}\left[r^k\cdot\sum_{i,j\in N}\sum_{f=1}^{F}\left|x_{ijf}^k\left(t_2\right)-x_{ijf}^k\left(t_1\right)\right|\right]+\sum_{k=1}^{K}\left[r^k\cdot\sum_{i,j\in N}\sum_{f=1}^{F}\left|x_{ijf}^k\left(t_3\right)-x_{ijf}^k\left(t_2\right)\right|\right]}{\left[\sum_{k=1}^{K}r^k\right]\cdot\bar{y}}\\
&= \frac{\sum_{k=1}^{K}\left[r^k\cdot\sum_{i,j\in N}\sum_{f=1}^{F}\left(\left|x_{ijf}^k\left(t_2\right)-x_{ijf}^k\left(t_1\right)\right|+\left|x_{ijf}^k\left(t_3\right)-x_{ijf}^k\left(t_2\right)\right|\right)\right]}{\left[\sum_{k=1}^{K}r^k\right]\cdot\left[\sum_{i,j\in N}\sum_{f=1}^{F}\frac{y_{ijf}\left(t_3\right)+y_{ijf}\left(t_1\right)}{2}\right]}\\
&\geq \frac{\sum_{k=1}^{K}\left[r^k\cdot\sum_{i,j\in N}\sum_{f=1}^{F}\left|x_{ijf}^k\left(t_3\right)-x_{ijf}^k\left(t_1\right)\right|\right]}{\left[\sum_{k=1}^{K}r^k\right]\cdot\left[\sum_{i,j\in N}\sum_{f=1}^{F}\frac{y_{ijf}\left(t_3\right)+y_{ijf}\left(t_1\right)}{2}\right]}\\
&= \Delta\left(\omega_{t_1},\omega_{t_3}\right)
\end{aligned}
$$

Figure 9: Triangle inequality derivation when average number of edges is constant

Start by considering $\Delta(\omega_{t_1},\omega_{t_2})+\Delta(\omega_{t_2},\omega_{t_3})$. Replace each Δ with its definition. The expressions in the denominator of both fractions for the average number of edges can be replaced by $\bar{y}$. Since both fractions now have a common denominator, they may be added to produce a single fraction. Now the summations in the numerator

are combined, and the $\bar{y}$ in the denominator is replaced with the average number of edges at times t_1 and t_3. Finally, since the triangle inequality holds for absolute value, replace the expression $\left|x_{ijf}^k(t_2) - x_{ijf}^k(t_1)\right| + \left|x_{ijf}^k(t_3) - x_{ijf}^k(t_2)\right|$ with $\left|x_{ijf}^k(t_3) - x_{ijf}^k(t_1)\right|$ by introducing the appropriate inequality. This produces an expression that is equivalent to $\Delta(\omega_{t_1}, \omega_{t_3})$. Thus, $\Delta(\omega_{t_1}, \omega_{t_2}) + \Delta(\omega_{t_2}, \omega_{t_3}) \geq \Delta(\omega_{t_1}, \omega_{t_3})$, as desired.

With a formula for Δ established, modifications to Erwin's formulation can now be explained. As mentioned in Chapter Two (II), Kleeman, et al. found that their heuristic occasionally returned a better solution than the "optimal" solution found using the MILP approach. This suggests that either they or Erwin had made an error. According to Kleeman, et al.:

> Since our stochastic method outperformed his solved deterministic methods, our results were surprising. After careful review of all objective functions and constraints, we found that our implementation was coded exactly as it was for Erwin. We determined that Erwin's black box implementation was the probable fault. His optimization algorithms do not have any parameters specifying the granularity of his commodity flows. His programmed limitations may have made it impossible for his deterministic algorithms to find the best solutions. [8:41]

In order to avoid incorporating the same error into this work, Erwin's code was re-implemented into Xpress-MP one constraint at a time. After each constraint was added, networks with known correct solutions were solved. Two errors were found using this approach. Occasionally, solutions were produced that routed commodities over edges that were specifically set to 0 in the potential-adjacency matrix A'. It was discovered that there was no constraint to enforce that x_{ijf}^k must be 0 whenever $a'_{ijf} = 0$. Thus, the constraint that $x_{ijf}^k \leq a'_{ijf}$ for all possible values of i, j, f, and k was added. The second error was found when larger input files began to be tested. This error generated

topologies that included edges in the design over some of which no commodities were routed. Since edges have a fixed cost, it does not make sense to include an edge unless it actually needs to be there. This secondary problem stems from the fact that the input files Erwin used were randomly generated. Random input files themselves are not a problem, if they are internally consistent. Erwin's input file generator allowed potential-adjacency matrices to have $a'_{ijf} = 1$ with $a'_{jif} = 0$. Since edges are required to be bidirectional, whenever a commodity is routed over (i, j, f), the edge (j, i, f) is forced to be included as well. Erwin had no constraint requiring $y_{ijf} \leq a'_{ijf}$. As a result, solutions were ending up more expensive than they were expected to be. Since there is a constraint that $x^k_{ijf} \leq y_{ijf}$ for all possible values of i, j, f, and k, adding the constraint that $y_{ijf} \leq a'_{ijf}$ for all possible values of i, j, f, and k fixes both this problem and the previous one as well. This additional constraint is reflected as inequality (2.3). There have been no additional errors detected.

With a corrected implementation, two ideas for how to approach modifying the algorithm to produce different topologies have been considered. The first method is to add new constraints to the MILP. The second technique is to add a penalty to the objective function for reusing edges. The first means is cumbersome, fraught with disadvantages, and had to be abandoned. In contrast, the second method is elegant and produces good results. The added constraints approach is briefly described before the penalty approach is given in full detail.

3.3.2 The Added Constraints Approach

When a solution is generated at time t_1, the values for $x_{ijf}^k(t_1)$, $y_{ijf}(t_1)$, and $m^k(t_1)$ are temporarily saved for comparison to the solution generated during the next iteration at time t_2. It is possible to use these saved values as constants in constraints. In this way, each iteration is solved with constraints of the same form, but having different coefficients. This results in different solutions produced for each iteration. The question is what constitutes a meaningful constraint in this situation.

One natural approach is to use the formula for Δ as a constraint. Specifically,

$$\Delta\left(\omega_{t_1}, \omega_{t_2}\right) \geq \Delta_{\min}. \tag{3.2}$$

Here, the variables corresponding to ω_{t_1} are held constant, leaving only the variables corresponding to ω_{t_2} truly variable. The $\Delta_{\min}$ on the right-hand side represents some minimum difference the new topology being generated must be from the previous topology.

There are several problems that using inequality (3.2) as a constraint introduces. First, (3.2) is not linear. If the definition for Δ contained a single absolute value, it may have been possible to split the single non-linear constraint into two linear constraints. Unfortunately, that is not possible with this inequality. At best, (3.2) can be made differentiable by replacing the absolute differences in Δ with squared differences. Even then, the result is a quadratic program. Quadratic programs can still be solved, but with no guarantee of optimality. Additionally, the extra difficulty of solving a quadratic program translates into much longer computation time.

The second (and more undesirable) problem is the way in which certain solutions can be found to satisfy the new constraint. In the original MILP, there is no constraint that specifically forbids routes from containing loops. A loop does not achieve anything besides costing extra money and consuming capacity that other commodities may need. Since the objective is to minimize cost, loops are naturally avoided. However, the addition of (3.2) forces more expensive solutions, and one way to achieve additional expense is by adding loops to a route. This defeats the purpose of polymorphic networking. An example of this problem is now demonstrated.

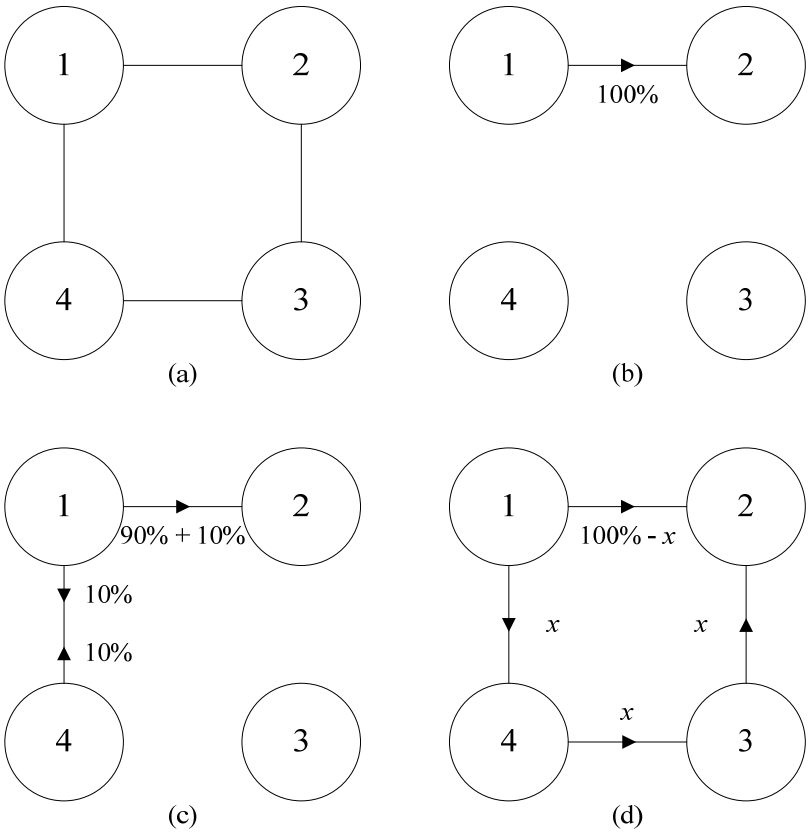

Figure 10: 4-node network showing looping problem

The four-node network illustrated in Figure 10 has nodes represented by circles with links shown as lines. When a particular direction is required, an arrowhead is placed

on the line. Figure 10(a) shows all the potential (bidirectional) edges of the network. Each edge has a fixed construction cost of 1 unit per direction and a routing cost of 1 unit/Kbps. Suppose there is a 1 Kbps commodity that needs to be routed from node 1 to node 2 and $\Delta_{min} = 0.0\bar{6}$. The first step is to find a good initial topology.

As shown in Figure 10(b), the optimal first solution involves routing 100% of the commodity directly from node 1 to node 2 (denoted 1→2). The total cost for this solution is 3 units (2 units for construction plus 1 unit for routing). Recall that even though traffic only flows one direction in this topology, both directions of the edge are included in the cost because of constraint (2.4). Due to the construction and routing costs involved, for the second solution, it is cheaper to send 10% of the commodity on the loop 1→4→1 before joining the remaining 90% over the path 1→2 as seen in Figure 10(c) than it is to send even a minuscule percentage x over the path 1→4→3→2 with the remainder over the path 1→2 as seen in Figure 10(d). The topology in Figure 10(c) has a total cost of 5.2 units, where just the construction cost for the topology in Figure 10(d) is 8 units. The Δ difference between Figure 10(b) and Figure 10(c) is $0.0\bar{6}$ and between Figure 10(b) and Figure 10(d) is $0.8x$. Thus, if $x = 8.\bar{3}\%$, then both Figure 10(c) and Figure 10(d) have the same measured Δ difference from Figure 10(b), but Figure 10(c) is always cheaper. Therefore, Figure 10(c) is given as the second solution using the added constraints approach. This is definitely an undesirable situation; it does not really alter the traffic flow that existed in Figure 10(b) in a useful way.

Various remedies to eliminate this problem were examined. A simple quadratic constraint that prohibits a commodity from flowing both directions on a single edge is

possible. A constraint requiring $x_{ijf}^k \cdot x_{jif}^k = 0$ for all possible values of i, j, f, and k is all that is needed. This is an expensive fix on account of the nonlinearity of the constraint and because there are a large number of edge/commodity pairs, even for small networks. Unfortunately, no such simple means for preventing less trivial, circular loops was found. Because of the quadratic constraint and no simple way to avoid having commodities sent over loops, the added constraints approach was deemed unsatisfactory. The alternative penalty approach is described next.

3.3.3 The Penalty Approach

Rather than introducing a new constraint to the MILP as described above, a different approach was examined where the routing costs are increased from iteration to iteration. This approach adds an extra cost to Erwin's objective function (2.1). The modified objective function is shown in (3.3). The b_{ijf}^k term indicates an additional cost for sending a commodity across an edge over which it has previously been sent. After each iteration, each b_{ijf}^k is incremented by the value found for x_{ijf}^k. In this way, the edge (i, j, f) is more costly for commodity k to use in the next iteration, and this encourages a new route to be found. The cost does not increase for commodities that were not previously routed over (i, j, f).

The polymorphic network problem (PNP) formulation is thus to periodically minimize

$$\sum_{(i,j,f)\in E}\left[\sum_{k=1}^{K}\left(v_{ijf}^k + b_{ijf}^k\right)x_{ijf}^k\right] + \sum_{(i,j,f)\in E} c_{ijf}\,y_{ijf} + \sum_{k=1}^{K} 1000 r^k m^k \qquad (3.3)$$

subject to

$$y_{ijf} = 0 \text{ or } 1 \text{ for all } (i, j, f) \in E, \tag{3.4}$$

$$y_{ijf} \le a'_{ijf} \text{ for all } (i, j, f) \in E, \tag{3.5}$$

$$y_{ijf} = y_{jif} \text{ for all } (i, j, f) \in E, \tag{3.6}$$

$$x^k_{ijf} \le y_{ijf} \text{ for all } (i, j, f) \in E, 1 \le k \le K, \tag{3.7}$$

$$x^k_{ijf} \ge 0 \text{ for all } (i, j, f) \in E, 1 \le k \le K, \tag{3.8}$$

$$\sum_{k=1}^{K} r^k x^k_{ijf} \le cap_{ijf} \text{ for all } (i, j, f) \in E, \tag{3.9}$$

$$\sum_{j \in N} y_{ijf} \le u_{if} \text{ for all } i \in N, 1 \le f \le F, \tag{3.10}$$

$$m^k = 0 \text{ or } 1 \text{ for all } 1 \le k \le K, \tag{3.11}$$

$$\sum_{\{j, f:(i,j,f) \in E\}} x^k_{ijf} - \sum_{\{j, f:(j,i,f) \in E\}} x^k_{jif} = \begin{cases} 1 - m^k & \text{if } i = s^k \\ m^k - 1 & \text{if } i = d^k \\ 0 & \text{else} \end{cases} \text{ for all } i \in N, 1 \le k \le K \tag{3.12}$$

where b^k_{ijf} is incremented by x^k_{ijf} for all possible values of i, j, f, and k after each solution is found.

The additional costs are kept in a separate matrix to allow for fair comparison between solutions. When a new solution is found, its cost based on the original price scheme can be calculated by subtracting the additional cost. The first solution found is optimal. Subsequent solutions are suboptimal or, at best, a different optimal solution, and this allows one to see how much extra they cost. The original formula for Δ is kept to measure the topological difference between solutions after they are generated.

The portion of the objective function that assigns penalties to dropped commodities is kept as it appears in Erwin's work [34:28]. The coefficient of 1000 is sufficiently large for the networks tested in this research. For a network of 40 nodes, the longest possible route for a commodity is 39 hops. Assuming an initial routing cost of 1 unit/hop with an increase of 1 unit/hop/iteration, after ten iterations, a commodity contributes at most $(39) \cdot (11) = 429$ units of cost to the network. It is always cheaper to route a commodity (if it fits) than it is to drop the commodity.

For larger networks, networks with more expensive routing costs, or applications that perform more than ten iterations, a more careful choice of penalty coefficient must be made. For future research, it is recommended that the 1000 coefficient be replaced with a more flexible term to ensure that the penalty for dropping a commodity remains more expensive than the cost of routing the commodity. Generalizing the analysis of the preceding paragraph, consider a network configuration consisting of n nodes. The longest possible route for a commodity is $n - 1$ hops. Let v_{max} be the maximum routing cost for any commodity over any edge. Finally, let L be the number of polymorphisms to be generated. Replacing 1000 with $(n - 1) \cdot (v_{max} + L)$ will keep the penalty sufficiently large. If L is not known in advance, then $(n - 1) \cdot (v_{max} + l)$ can be used, where l is the number of the polymorphism currently being solved.

In practice, as more and more iterations are solved, the routing costs involved grow linearly. In a 64-bit machine, there is little danger of the costs generating an overflow. If this becomes a concern, a scaling factor can be applied to the b_{ijf}^{k} terms. However, it is likely that the networking conditions used as an input to the polymorphic

networking algorithm will eventually change. When this occurs, the algorithm will be restarted with new inputs and the b_{ijf}^{k} terms reset back to 0.

In the process of working with the now-abandoned additional constraints approach, the choice was made to move the implementation of the PNP formulation from Xpress-MP into General Algebraic Modeling System (GAMS). GAMS is a "high-level modeling system for mathematical programming and optimization" [59]. GAMS allows for representation of a model which it then translates into a form to be solved by various solvers such as BDMLP for linear models and LaGO or CONOPT for nonlinear models. The original impetus for the switch was GAMS's ability to handle quadratic constraints. See Appendix G for a sample of the GAMS model code used for solving the PNP. Appendix H shows the results for a simple network of five nodes, each with four interfaces, and five commodities.

The inputs necessary for the calculations in this formulation include the potential-adjacency matrix; the source, destination, and bandwidth requirement for all the commodities; and both the fixed costs and variable costs for all the possible edges. It is important to know how the values of these inputs can be obtained to be placed into a network description file for the algorithm.

In a non-mobile network, the set of potential edges can generally be assumed to be constant. That is, the edges already exist. Inclusion in a topology simply amounts to whether any traffic is being sent over the edge or whether the edge is kept idle. In this case, the fixed cost does not equate to actual construction cost. Nonetheless, the cost is

still useful as a means for keeping the number of edges in the topology at a minimum. The result is a network with more highly utilized links and fewer underutilized links.

In mobile networks, the set of potential edges is expected to be in constant flux. It is entirely likely that the edges that are considered at the time the calculations begin may no longer exist upon completion of the calculation, making it impossible to implement the topology. In order for polymorphic networking to work well in a mobile environment, there needs to be some means of knowing in advance which nodes will be involved, where they will be, and when they will be there. These are exactly the details that the NTO process is designed to bring to the forefront.

The source, destination, and bandwidth requirement for all commodities involved is more scrutable. In a fixed network, individual nodes can be set up to send requests to a centralized processor responsible for generating the polymorphisms. Each request specifies the desired destination and bandwidth required. Similar requests can also indicate when a commodity is no longer needed. Another approach, if traffic patterns are stable, is to monitor the network to compile a list of who talks to whom and the amount of bandwidth utilized. In a military setting, the NTO process is again proposed as a means for knowing this type of information in advance.

The fixed and variable costs can be defined in various ways. The cost need not be monetary, but perhaps related to the security of a link or the power required to operate a link. If a particular edge is believed to be compromised, its cost can be artificially raised to make it less likely to be picked for inclusion in a topology (or it can be removed from the potential-adjacency matrix to avoid having it used at all). If a particular interface type

requires more power, or time to establish, the cost of edges using that interface can be given higher cost to cause them to be used less frequently.

For rigorous testing, GAMS needs to be able to run solvers on a large variety of network types. The input files have been designed to be as flexible as possible for this reason. As a result, even for a small network of five nodes, it can take over an hour to create the input files by hand. A file generator has been written that can quickly generate the network description files needed as input for the GAMS program. Currently, the file generator spawns network characteristics randomly, within prescribed ranges.

There are many variables that can be set for the input files. To avoid an astronomical number of test configurations, the variables are limited to the number of nodes, the number of interface types available, and the number of commodities each node sends. Networks of 5, 10, 15, 20, 25, 30, 35, and 40 nodes are considered. Each node has the same number of interface types with the maximum number of each interface type set at 4. Networks with 1, 2, 3, and 4 interface types are examined. Every node is the source of 1, 2, or 3 commodities, with the destinations randomly assigned. See Table 1 for the enumeration of test configurations. For each choice of number of nodes, there are 12 combinations of number of commodities and interface types. Thus, there are 96 configurations to test. Shorthand for a particular configuration is given by #N#C#I, where each of the '#' symbols is replaced by a value from Table 1 below. For example, 10N3C2I denotes the configuration with 10 nodes, 3 commodities per node, and 2 interface types per node.

Table 1: Test configurations for polymorphic networking

Nodes	5	10	15	20	25	30	35	40
Commodities per Node	1	1	1	1	1	1	1	1
	2	2	2	2	2	2	2	2
	3	3	3	3	3	3	3	3
Interface Types per Node	1	1	1	1	1	1	1	1
	2	2	2	2	2	2	2	2
	3	3	3	3	3	3	3	3
	4	4	4	4	4	4	4	4

All edges have their construction cost set at 1 unit. The routing cost for every edge and every commodity is initially set to 1 unit for 100% flow. The routing costs increase from iteration to iteration through the b_{ijf}^{k} parameter. Every commodity has a required bandwidth of 10 Kbps, and every edge has a capacity of 100 Kbps. It is not desired that any commodities be dropped due to lack of capacity.

The potential-adjacency matrix is randomly generated with 0's along the diagonal (no edges from a node to itself over any interface). The entries above and below the diagonal are symmetric. Symmetry is important because edges are required to be bidirectional. If an edge from node i to node j on interface f is available, then so too must the edge from node j to node i on interface f be available. Adjacency is set to a percentage based on the number on nodes. As the number of nodes n increases, the number of possible edges increases with n^2. To keep the number of edges per node consistent across configurations, the adjacency percentages are set at $9/[4(n-1)]$. This formula was chosen to result in 25% adjacency for a 10-node network. Table 2 shows the corresponding percentages across the number of nodes tested. With this approach, the average number of potential edges per node is constant with respect to the number of

nodes, but increases with respect to the number of interface types. At the percentages listed, the average number of potential edges per node is $2.25F$, where F is the number of interface types.

Note that the adjacencies are potential. A solution to the PNP includes, in general, far fewer edges than what is available. One reason is that nodes have degree limitations. Another reason is that a subset of edges may be all that is needed to fully accommodate all the commodities that need to be routed.

Table 2: Adjacency percentage as a function of number of nodes

# of Nodes	Adjacency (%)
5	56.25
10	25.00
15	16.07
20	11.84
25	9.38
30	7.76
35	6.62
40	5.77

Since the potential-adjacency matrix and the destinations of commodities are randomly generated, multiple input files are used for each configuration so that confidence intervals can be determined. For configurations from 5 to 20 nodes, 30 input files are used. That is, there are 30 test cases for each configuration. For larger configurations, the number of test cases considered is limited due to the extreme time required to solve. Each potential-adjacency matrix is tested for connectedness prior to running the GAMS program.

The particular solver chosen for these experiments is CoinCbc 2.2. Out of the licensed linear solvers available, CoinCbc 2.2 was able to handle the large number of

constraints and variables required, and found optimal solutions in the shortest amount of time. For each test case, GAMS is programmed to run CoinCbc through 10 iterations to generate 10 polymorphisms. The cost of each solution along with its measured distance from the previous solution and the time to solve are tabulated. In addition, the metrics of network diameter and average number of hops are kept. The analyses and results for these test cases are found in Chapter Four (IV).

A sample network with five nodes and two interface types is presented to illustrate the results of the penalty approach to polymorphic networking. A network of small size is chosen for clarity of explanation. Figure 11 shows the network with the full set of potential edges. The nodes are labeled 1 through 5. Solid lines indicate a potential connection over interface 1 and dashed lines indicate a potential connection over interface 2. The edge adjacencies are randomly generated at 56.25%. Notice that there does not exist an edge for both interface types between all possible pairs of nodes. All edges have a fixed construction cost of 1 unit and a capacity of 100 Kbps.

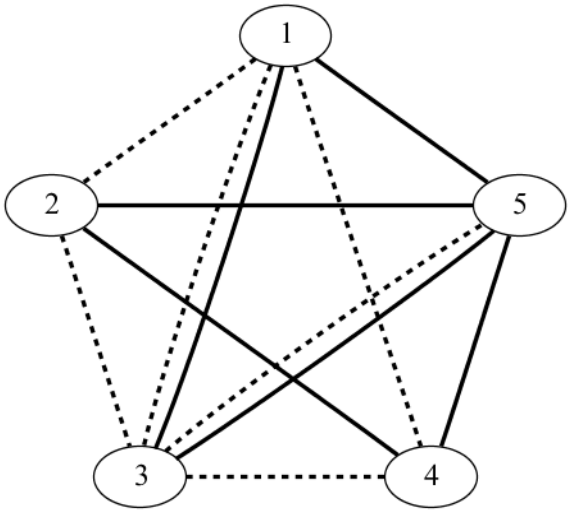

Figure 11: 5-node example showing all potential edges in network

The set of commodities that must be routed for this network are given in Table 3. Each node is the source for one commodity and the destinations are randomly generated. Every commodity has a bandwidth requirement of 10 Kbps. For every commodity, the cost to send 100% of the required bandwidth across any edge is initially set to 1 unit.

Table 3: Description of commodities for 5-node example

Commodity	Source	Destination	Bandwidth
1	1	5	10 Kbps
2	2	4	10 Kbps
3	3	5	10 Kbps
4	4	1	10 Kbps
5	5	4	10 Kbps

The first four (of ten) polymorphisms generated by GAMS/CoinCbc are shown in Figure 12(a)-(d). Note that all edges chosen for these topologies are among the potential edges shown in Figure 11. Also, not every potential edge shown in Figure 11 is among the edges chosen for these four polymorphisms.

In Figure 12(a), eight edges are used for a construction cost of 8 units. All commodities, save commodity 4, have a single-hop path to follow. Commodity 4 must follow two hops – one from node 4 to node 5 and one from node 5 to node 1. Thus, the routing cost is 6 units for a total cost of 14 units. Similarly, the topology in Figure 12(b) has a construction cost of 8 units. The routing cost for the second topology is a little less straightforward. A total of seven hops are taken by the five commodities. However, the edge used by commodity 5 is the same edge as in the previous topology. Hence, the routing cost on this edge for commodity 5 has increased to 2 units. Therefore, the combined routing cost is 8 units, for a total cost of 16 units. Since the increased cost for commodity 5 is added artificially to encourage distinct solutions, the extra 1 unit can be

subtracted to yield a true cost of 15 units. The cost information for all ten polymorphisms is given in Table 4.

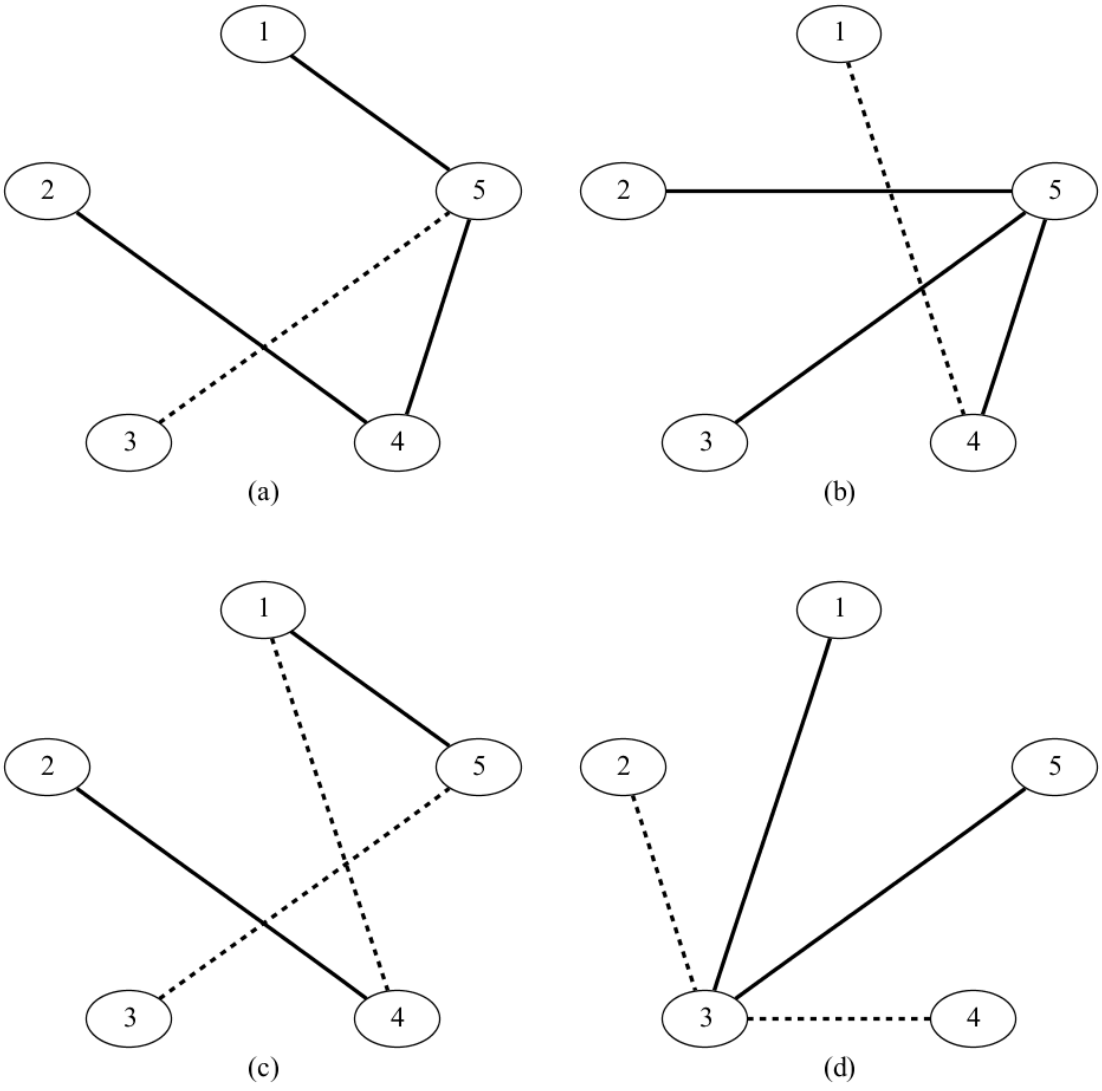

Figure 12: First four polymorphisms for 5-node example

Table 4: Cost information for all 10 polymorphisms of 5-node example

Polymorphism	Construction Cost	Routing Cost	Total Cost	True Cost
0	8	6	14	14
1	8	8	16	15
2	8	10	18	14
3	8	10	18	17
4	8	11	19	17
5	8	14	22	16
6	8	15	23	17
7	8	17	25	16
8	8	19	27	16
9	8	19	27	16

As this is a small example, the time required to find each of the ten polymorphisms is very short. The mean time is 0.177 seconds, with a minimum of 0.081 seconds, a maximum of 0.63 seconds, a median of 0.119 seconds, and a standard deviation of 0.164 seconds. A plot of the solution times is given in Figure 13. Clearly, polymorphism 2 is an outlier. This suggests that the median may be a more telling statistic than the mean.

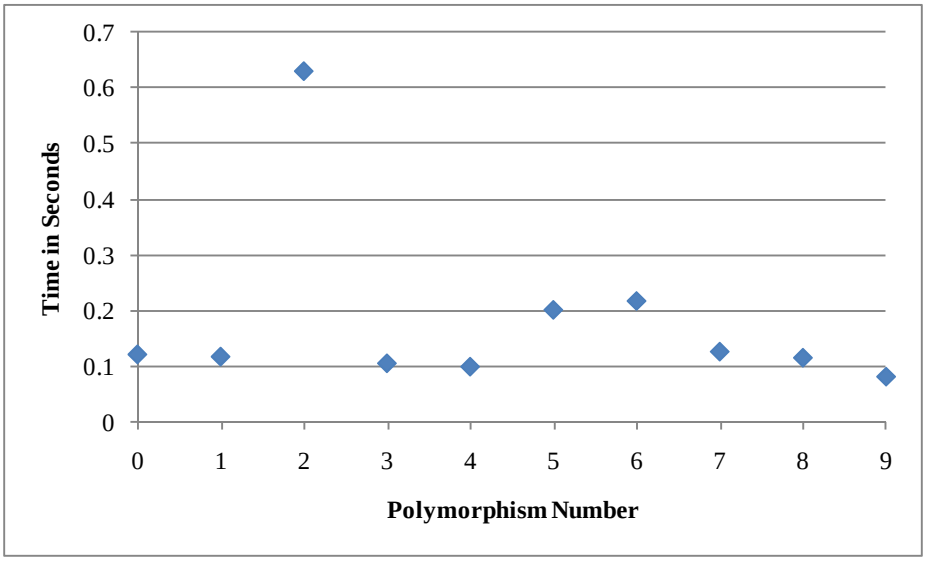

Figure 13: Solution times for 10 polymorphisms of 5-node example

Figure 14 is a plot of the median solution times of ten polymorphisms for all thirty randomly generated test cases. The average of the median values is 0.122 seconds. The network described in Figure 11 is actually case 2 on this plot.

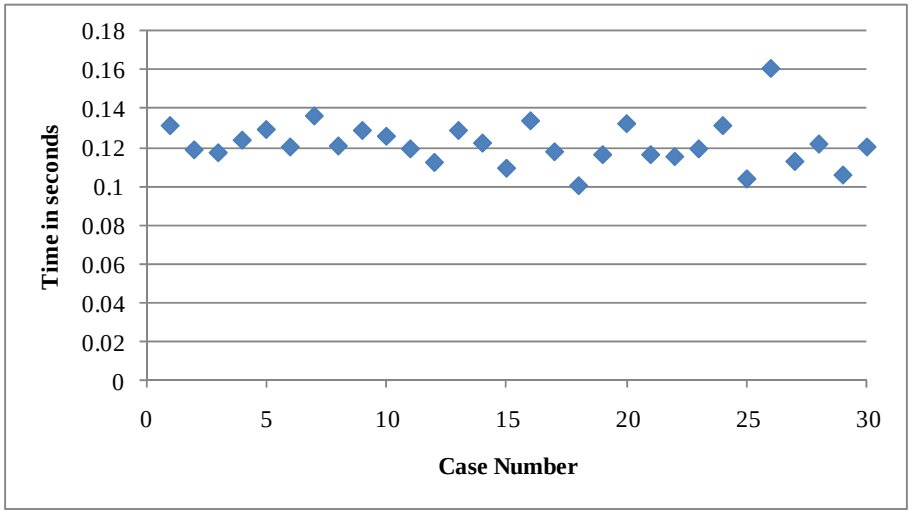

Figure 14: Median solution times for 30 cases of 5-node example

Finally, the measured difference between polymorphisms using Δ is now given. Table 5 lists these measured differences. Polymorphism 0 does not have a previous topology to measure against, so its entry is left blank. Note that no polymorphism is identical to the previous solution. In fact, perusal of all ten solutions shows them to be all distinct for this case. This is not true in general. Depending on the initial adjacency matrix, there can be arrangements where a handful of solutions are cycled or where only a single solution exists.

Table 5: Measured topological differences in 5-node example using Δ formula

Polymorphism	0	1	2	3	4	5	6	7	8	9
Δ		0.275	0.275	0.375	0.45	0.425	0.425	0.425	0.4	0.4

To help make formula (3.1) more clear, the calculation of Δ for polymorphism 1 is now given. Since there are five commodities and two interface types, $K = 5$ and $F = 2$. Also, $r^k = 1$ Kbps for $k = 1, \ldots, 5$. Table 6 lists the nonzero variables $x_{ijf}^k(t)$ and $y_{ijf}(t)$ for $t = 0$ and $t = 1$. Since the y_{ijf} variables are binary, nonzero values must equal 1. The values for the x_{ijf}^k variables are percentages and must be from the interval [0,1]. In this instance, no commodity is being split among multiple paths, so all the nonzero values are again equal to 1. In the numerator of formula (3.1), all absolute differences are 0 except for those involving the x_{ijf}^k in Table 6. Of those, only x_{541}^5 appears both for $t = 0$ and $t = 1$. Thus, the numerator of (3.1) evaluates to 11. From Table 6, the average number of used edges in the network at $t = 0$ and $t = 1$ is 8. Therefore, formula (3.1) reduces to $11/(5 \cdot 8) = 0.275$.

Table 6: Nonzero variables for computing Δ for polymorphism 1 of 5-node example

	$t = 0$	$t = 1$
Nonzero $x_{ijf}^k(t)$	$x_{151}^1, x_{241}^2, x_{352}^3, x_{451}^4, x_{511}^4, x_{541}^5$	$x_{142}^1, x_{451}^1, x_{251}^2, x_{541}^2, x_{351}^3, x_{412}^4, x_{541}^5$
Nonzero $y_{ijf}(t)$	$y_{151}, y_{241}, y_{352}, y_{421}, y_{451}, y_{511}, y_{532}, y_{541}$	$y_{142}, y_{251}, y_{351}, y_{412}, y_{451}, y_{521}, y_{531}, y_{541}$

3.4 Measuring the Security of Polymorphic Networks

Because of time and cost limitations, physical networks that can embody the polymorphisms generated by the PNP formulation described in the previous section have not been built. As a result, live testing of resistance to various cyber-attacks is not yet possible. However, there is a measurement that can be made to provide a sense of

potential benefit. Consider a static network. Presumably, if an attacker has found an active link over which to eavesdrop, that link remains active allowing the attacker to gather a continuous stream of data from whatever commodity may be utilizing that link. With a polymorphic network, on the other hand, each link may be active during some polymorphisms and idle during others. For example, Figure 12 in the previous section shows edge $(1,5,1)$ to be active in (a) and (c), but idle in (b) and (d). Hence, an attacker listening on this link over this time period can intercept a maximum of 50% of commodity 1. In fact, if all ten polymorphisms are examined, the average percentage active time (APAT) for the edges of the network in Figure 11 is 33.33% with a standard deviation of 7.61%. Looking at all thirty randomly generated test cases, the average active time goes down to 33.19% with a standard deviation of 8.73%. In all these cases, no edge is active more than 60% of the time. In these calculations, edges that are never utilized do not contribute to the average.

For analysis, each of the test configurations generated for testing the penalty approach for the PNP is examined for APAT. An attempt is made to identify trends in APAT values with respect to increasing the number of nodes, the number of interfaces/node, and the number of commodities/node. In addition to APAT, each configuration is also tested for edges that are active 100% of the time. Such an edge provides an attacker with a persistent source of data, albeit data that potentially comes from different commodities. It is expected that the smaller a network is in terms of options for routing a commodity, the greater the potential for edges that must be included in all ten polymorphisms.

3.5 Summary

This chapter detailed the methodology and approach used during the course of this research. Each of the four research objectives were discussed in turn. First, the NTO process was developed and described in the setting of a JAOC. The flow of data for the NTO begins with planning documents such as the ATO, historical precedent, and a capabilities database. The conglomeration (as pre-NTO) is analyzed with feedback to other planning teams, and individual taskings are published in a finalized NTO. An example of a mission in the ATO resulting in a corresponding NTO tasking was provided. Next, three scenarios whose intent is to show how the existence of an NTO process can improve the QoS of the Global Information Grid (GIG) were introduced. Afterward, a deterministic approach to the PNP based on Erwin's MILP formulation for the MCNDP was developed. A semimetric was defined to measure the difference between network topologies. During development, deficiencies in Erwin's MCNDP formulation were found and corrected. Two approaches to the PNP were described: an added constraints approach and a penalty approach. A simple 5-node example was presented to illustrate the results of the penalty approach to polymorphic networking. Also, the plan for testing the formulation using the GAMS modeling system was explained. The chapter concluded with a discussion of how examining the APAT for edges in a polymorphic network can give a measure of the increased resistance of a network to cyber attack. The next chapter presents analysis and results from the three scenarios and the penalty approach for the PNP.

IV. Analysis and Results

This chapter is split into several sections. The first section provides the details of the three scenarios introduced in Chapter Three (III) that were designed to show that following a Network Tasking Order (NTO) process can improve the quality of service (QoS) of the Global Information Grid (GIG). The analysis and results from the simulations implementing the scenarios are given. Chapter Three (III) also described a novel approach to polymorphic networking. A polymorphic networking problem (PNP) formulation using a mixed-integer linear program (MILP) approach was presented. This approach has been implemented and tested under a variety of configurations. The figures from these tests are also given in the second section of this chapter. The third section addresses the security benefits that implementing a polymorphic network provides in terms of the Average Percentage Active Time (APAT).

4.1 NTO Scenarios

4.1.1 NTO Scenario 1

The first scenario consists of two individual sources generating information that needs to be sent to a common headquarters (HQ). HQ is far enough away that direct communication from the two sources is not available, but there is an intermediate node that can act as a router. Both sources have a 36 Kbps connection to the router, and the router has a 36 Kbps connection to HQ. The router has a first in, first out queue with a buffer capacity of 50 packets. Source 1 (S1), an MQ-1 Predator Unmanned Aerial System (UAS), produces a high-priority video feed at a rate of 30.6 Kbps over a 30-minute time span that falls between the hours of 1400 and 1500. Source 2 (S2), a sensor net, produces

a continuous stream of data at a rate of 7.2 Kbps. S2's data is of lower priority and not time-sensitive, as it is collated at HQ and reviewed once daily. In addition to the router, there is also an E-3 Sentry Airborne Warning and Control System (AWACS) aircraft flying a 30-minute orbit during the hours from 1300 to 1800 in the airspace between S2 and HQ. This particular AWACS is known to be carrying equipment that allows wireless line-of-sight (LOS) networking at a rate of 28.8 Kbps. The orbit is such that it is never in LOS of S1. It is within LOS of both S2 and HQ for 10 minutes each, with no overlap in these time spans. Thus, the aircraft cannot act as a router. However, the aircraft can be used as a data ferry, storing the information that is uploaded from S2 and downloading it later to HQ when it is in range. An overhead view of the scenario is given in Figure 15.

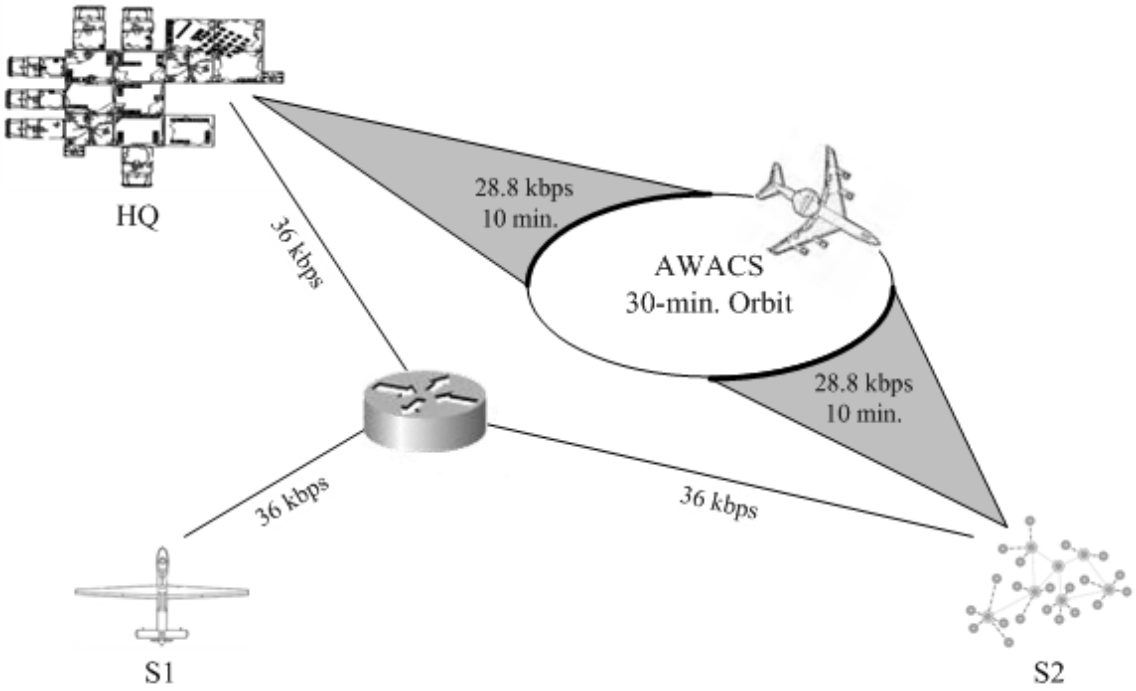

Figure 15: Scenario 1 overview

This scenario is first considered without an NTO process in place for guidance. S1's operators do not have any options for routing the video feed; they must utilize the router. S2's operators have a little more flexibility. Given the choice between sending its traffic to the router or to the AWACS, the operators are likely to send their traffic through the router over the 36 Kbps link. This seems like a reasonable choice given the larger bandwidth of this route and the delays that are associated with data ferrying. It is probable that S2's operators are unaware that there is a 30-minute period where S1 is also sending data through the router. Even if S2's operators were aware of S1's mission, they are not privy to S1's data path to HQ. Further, they are not aware of the router's capabilities or the bandwidth of its connection to HQ. As a final point, without an NTO, the router itself has no way of knowing the relative priorities of messages coming from S1 and S2. Even if the router is using a protocol such as Differentiated Services [60], S1 and S2 have not been directed what forwarding behaviors to put into their packet headers.

Next, the same scenario is envisioned with an NTO process in place, enhancing *GIG-awareness*. In the planning pre-NTO phase, analysts expose the potential for the combined data rates from S1 and S2 to overwhelm the router's queue. The time span from 1400 to 1500 is deemed to be a contention period. At this point, there are multiple courses of action available.

1. Increase the bandwidth on the links through the router.
2. Deploy a second router.
3. Turn off the data flow from S2 (neither store nor send) during the contention period.
4. Use the AWACS as a data ferry during the contention period.

5. Give the router a priority queue and mark the packets from S1 and S2 with relative priorities.

Options 1, 2, and 5 are rejected because they require a physical change to or addition of equipment. Option 3 is rejected because, even though the data is of lower priority, it is not desired to lose the data. Since it is known that the information from S2 is not time-sensitive, the delay associated with option 4 is tolerable. Also, no changes need to be made to the other planning documents. This course of action approval is made at the operation level and given as directives to the operators of S2 and of the AWACS through the NTO. One directive orders S2 to send transmissions to the AWACS from 1400-1500. The second tasking orders the AWACS to receive those transmissions and deliver them to HQ. See Figure 16 for example NTO excerpts indicating how these taskings may appear. The increase in end-to-end delay for S2 is justified by the increased reliability of the overall network and the lower priority of S2's data. Because the locations of the AWACS and S2 are known, this information can be included in the NTO to assist in prepositioning of antennas, if necessary.

```
TSKCNTRY/US//
SVCTASK/A//
TASKUNIT/15AMD/ICAO:KFFO//
TASKNODE/SNTYP:MSCS/SNET 4//
1RTEDAT
/DEST     /START     /STOP      /NHOP            /LOC
/ICAO:HQ01/241400ZAPR/241500ZAPR/CALL:SKYWATCH 26/TRACK12//
GENTEXT/SKYWATCH 26 TO DATA FERRY//
SVCTASK/F//
TASKUNIT/552ACW/ICAO:KNFA//
TASKNODE/ACTYP:E3/SKYWATCH 26//
2FRYDAT
/SOURCE     /LOC   /START     /STOP      /FRYTO    /LOC
/CALL:SNET 4/BLUE13/241400ZAPR/241500ZAPR/ICAO:HQ01/BLUE23//
```

Figure 16: NTO excerpts for Scenario 1

This scenario is executed in the open source ns-2 simulator (version 2.29) [61]. Traffic is generated with constant bit rate generators using User Datagram Protocol (UDP). The data rates at which S1 and S2 send packets can be achieved in many ways by adjusting packet size and the interval between packets. One extreme way is to have S1 send a 3,825-byte packet once per second and for S2 to send a 900-byte packet once per second. To make the simulation more realistic, packet sizes of 8 bytes through 64 bytes in steps of 8 bytes are considered for both sources. The time interval between packets is adjusted accordingly. Random jitter is added to the traffic generators to prevent having consistently simultaneous arrivals at the router of packets from the two sources. Because of the random nature of the traffic, results are gathered using 30 different random seeds. This necessitates 1,920 simulation runs each for the two versions of this scenario (without and with the NTO). The various connections are implemented as simplex-links[7] with 5 millisecond (ms) delays and DropTail[8] queues (with default buffer size). No noise or signal fading is simulated. Only the one-hour contention period from 1400 to 1500 is simulated.

For the scenario with no NTO process in place (S2 using the router), S2 generates traffic for an hour, and S1 generates traffic for the 30-minute span from 1410 to 1440. For the scenario with an NTO process in place (S2 using the AWACS), the timing is more complicated. S2 needs to send 30-minutes worth of data (12.96 Mbits) within a 10-minute window of opportunity. To allow some time for connections to be established between S2 and the AWACS, it is arranged for the data to be sent in only 8 minutes.

[7] A simplex-link is a point-to-point connection over which signals flow in only one direction.
[8] DropTail queues employ a first in, first out service discipline and drop-on-overflow buffer management.

Consequently, the 12.96 Mbits of information in 8 minutes corresponds to an increased rate of 27 Kbps. The worst case for end-to-end delay happens when the LOS contact between HQ and the AWACS ends just prior to when the LOS contact between S2 and the AWACS begins. This is the situation modeled. S1 generates traffic for the 30-minute span from 1410 to 1440. S2 sends traffic to the AWACS in two 8-minute intervals from 1400 to 1408 and from 1430 to 1438. The AWACS relays the traffic to HQ in two 8-minute intervals from 1422 to 1430 and from 1452 to 1460.

The interarrival times (in ms) for packet generation at S1 and S2 for the various packet sizes are shown in Table 7. Two rows of interarrival times are given for S2 for the different data rates used in the scenario without and with the NTO. The interarrival times differ with packet size to keep the data rates constant.

Table 7: Packet interarrival times for Scenario 1

	Packet Size (bytes)							
	8	**16**	**24**	**32**	**40**	**48**	**56**	**64**
S1 interarrival time (ms) 30.6 Kbps	2.09	4.18	6.27	8.37	10.46	12.55	14.64	16.73
S2 interarrival time (ms) 7.2 Kbps (without NTO)	8.89	17.78	26.67	35.56	44.44	53.33	62.22	71.11
S2 interarrival time (ms) 27 Kbps (with NTO)	2.37	4.74	7.11	9.48	11.85	14.22	16.59	18.96

The results for Scenario 1 without an NTO process in place are examined first. In Table 8, the mean percentages of packets from S1 that get dropped at the router are shown. The table is arranged by the size of packets originating from both sources. Cells are shaded based upon their value. The higher the percentage, the darker a cell is shaded. S1 experiences less loss when packet sizes are relatively similar. S1 suffers the least loss

of 3.4518% when S1 packets are 40 bytes and S2 packets are 56 bytes. The largest loss of 4.7082% occurs when S1 packets are 32 bytes and S2 packets are 8 bytes.

Table 8: Mean % of S1 packets dropped in Scenario 1 (no NTO)

Source 1 Packet Size (bytes) \ Source 2 Packet Size (bytes)	8	16	24	32	40	48	56	64
8	3.51	3.56	3.78	3.95	4.05	4.04	4.42	4.40
16	4.44	3.51	3.48	3.57	3.68	3.78	3.85	3.96
24	4.67	4.18	3.51	3.46	3.49	3.56	3.65	3.71
32	4.71	4.42	4.02	3.51	3.46	3.47	3.51	3.56
40	4.68	4.59	4.29	3.92	3.50	3.46	3.45	3.49
48	4.64	4.67	4.43	4.17	3.86	3.50	3.46	3.46
56	4.63	4.69	4.56	4.33	4.07	3.80	3.50	3.45
64	4.67	4.70	4.63	4.44	4.23	4.01	3.74	3.49

In Table 9, the mean percentages of packets from S2 that get dropped at the router are shown. As before, the table is arranged by size of packets originating from both sources, and cells are shaded based upon their value. In contrast to S1, S2 experiences more loss when S1 and S2 packet sizes are relatively similar. S2 suffers the least loss of 2.4988% when S1 packets are 64 bytes and S2 packets are 16 bytes. The largest loss of 5.1471% occurs when S1 packets are 16 bytes and S2 packets are 24 bytes.

The confidence intervals (CI) for the mean percentages shown in Table 8 and Table 9 are very tight. Figure 17 shows the 95% CI for the mean percentage of S1 packets dropped at the router when S1 packets are fixed at 56 bytes and packets from S2 range in size from 8 bytes to 64 bytes. Figure 18 shows the 95% CI for the mean

percentage of S2 packets dropped at the router when S2 packets are fixed at 56 bytes and packets from S1 range in size from 8 bytes to 64 bytes. These plots are typical of the CI for all mixtures of packet sizes. Plots of all CI results can be found in Appendix I.

Table 9: Mean % of S2 packets dropped in Scenario 1 (no NTO)

Source 1 Packet Size (bytes) \ Source 2 Packet Size (bytes)	8	16	24	32	40	48	56	64
8	5.07	4.94	4.48	4.11	3.91	3.91	3.08	3.17
16	3.08	5.07	5.15	4.93	4.70	4.47	4.35	4.09
24	2.60	3.62	5.06	5.15	5.06	4.92	4.73	4.61
32	2.51	3.08	3.95	5.04	5.14	5.13	5.03	4.91
40	2.59	2.73	3.37	4.16	5.06	5.13	5.15	5.07
48	2.66	2.58	3.09	3.63	4.28	5.02	5.09	5.12
56	2.64	2.52	2.83	3.27	3.82	4.39	5.00	5.09
64	2.58	2.50	2.65	3.07	3.46	3.93	4.47	5.00

Note that for no combination of packet sizes does either source experience an acceptable amount of loss. The QoS is degraded significantly for both sources. The high-priority video feed losses from S1 are especially troubling. The suggested tolerance for data loss for high-quality audio-video streaming is below 1%, and for two-way interactive audiovisual services it is below 2-3% [62:40]. These tolerances are all surpassed for S1's data stream.

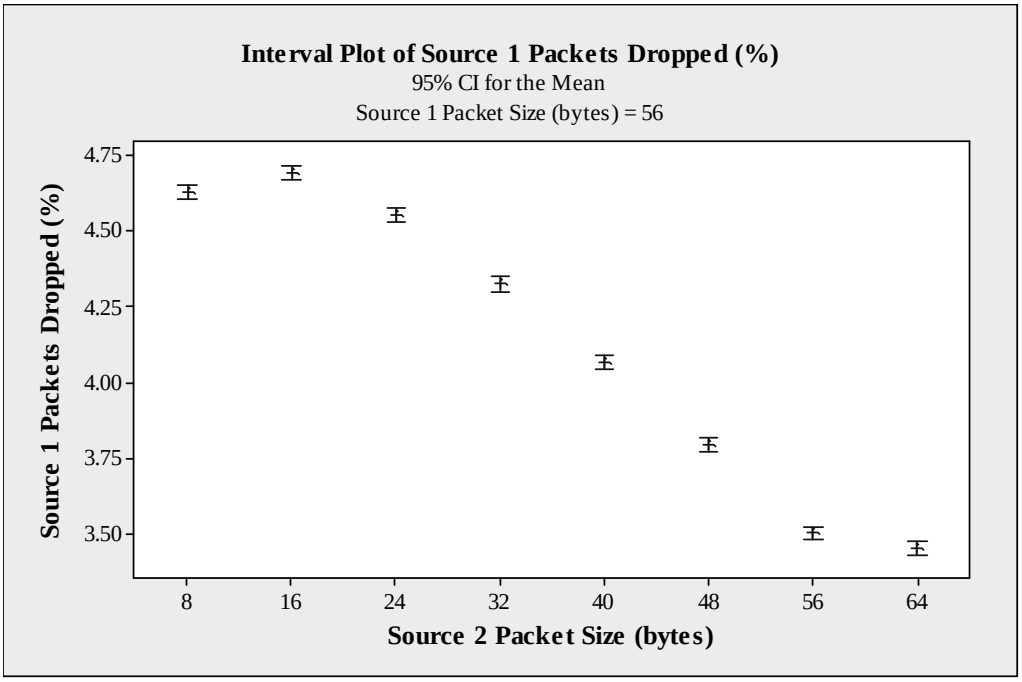

Figure 17: 95% CI for mean % of 56-B S1 packets dropped in Scenario 1 (no NTO)

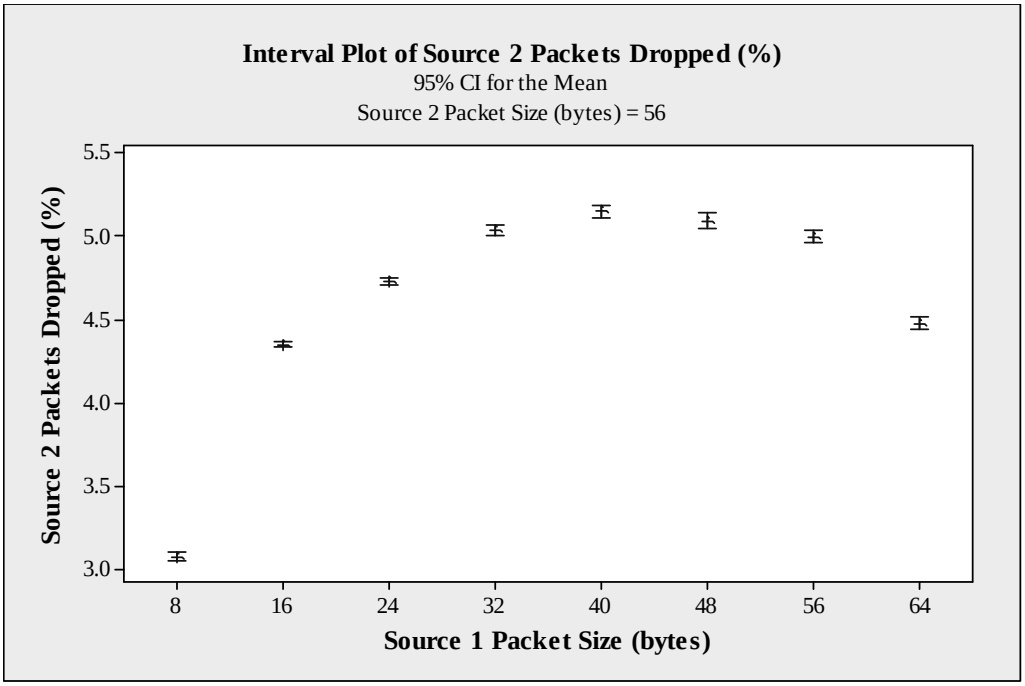

Figure 18: 95% CI for mean % of 56-B S2 packets dropped in Scenario 1 (no NTO)

The User Datagram Protocol used in the simulation does not retransmit lost packets. Therefore, it is of interest to see what the total loss from both sources is in terms of bytes. Table 10 shows the mean percentages of total bytes lost from both sources at the router, broken down by packet sizes. The cells are again shaded based on their value, but the range of values is rather narrow. Percentages range from 3.9723% to 4.0116% with a trend of higher percentages occurring when packet sizes are small and lower percentages occurring when packet sizes are larger. Figure 19 provides a summary of statistics on the percentage of total bytes dropped at the router over all 1,920 simulation runs. The overall average is a loss of 3.9967% of the total bytes sent, with a very narrow 95% CI of 3.9947% to 3.9986%.

Table 10: Mean % of total bytes dropped in Scenario 1 (no NTO)

		Source 2 Packet Size (bytes)							
		8	**16**	**24**	**32**	**40**	**48**	**56**	**64**
Source 1 Packet Size (bytes)	**8**	4.0102	4.0062	4.0054	4.0046	4.0041	3.9996	3.9911	4.0039
	16	4.0032	4.0089	4.0116	4.0028	4.0032	3.9984	4.0073	4.0003
	24	4.0057	4.0018	4.0018	3.9970	3.9946	3.9937	3.9921	3.9958
	32	4.0036	3.9960	3.9997	3.9992	3.9974	3.9982	3.9964	3.9895
	40	4.0080	3.9910	3.9978	3.9992	3.9992	3.9955	3.9939	3.9956
	48	4.0028	3.9993	4.0031	3.9984	3.9919	3.9843	3.9844	3.9892
	56	3.9962	3.9977	4.0053	3.9884	3.9894	3.9891	3.9826	3.9767
	64	3.9972	3.9973	3.9976	3.9991	3.9834	3.9819	3.9750	3.9723

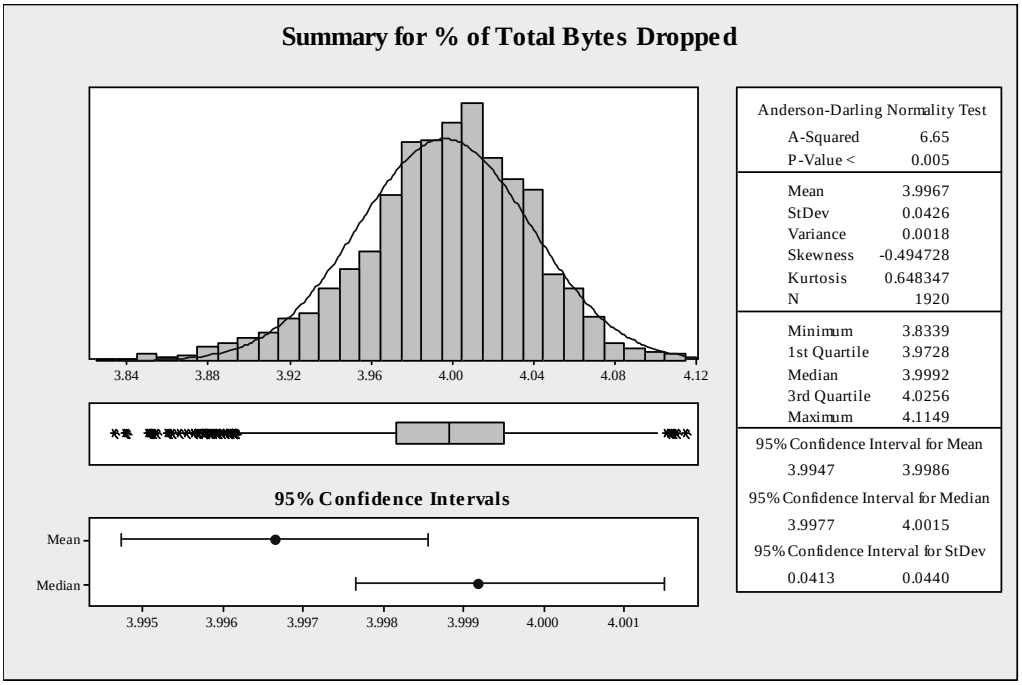

Figure 19: Summary statistics for % of total bytes dropped in Scenario 1 (no NTO)

The results from the simulations run for Scenario 1 with an NTO are very simple. No packets were lost from either source, regardless of the mix of packet sizes or random seed. Also, all data sent from S2 to the AWACS was subsequently sent from the AWACS to HQ. Bear in mind that no noise or signal fading was simulated; however, loss from those sources mainly applies to the lower priority S2 data.

Scenario 1, while simple and intuitive, is realistic and shows how the NTO can really make a difference. Keep in mind that the intuitiveness of this scenario is misleading. The scenario was described in great detail in Chapter Three (III), including an overhead picture of the situation. This description is basically a pre-NTO in narrative format. The equipment involved along with their missions, locations, and time periods is information that comes from planning documents such as the ATO. The capabilities database provides the figures on connection speeds, ferrying ability, and queue capacity.

Historical precedent gives the data on communication patterns. All of the information needed to understand the scenario is conveniently put together in one place. This makes it clear that there is a potential conflict, but in current practice, no such convenience exists.

4.1.2 NTO Scenario 2

The second scenario utilizes a derivative of a Combat Search and Rescue (CSAR) scenario developed by the MITRE Corporation for testing the suitability of the Joint Tactical Radio System (JTRS) Wideband Networking Waveform (WNW) to support future operations [63:1-1]. MITRE chose the CSAR scenario as it is "small enough in scale to readily model, yet diverse enough in mission types and participants involved to provide a realistic test of the WNW abilities" [63:iii]. The MITRE scenario involves 63 nodes; including fixed, land mobile, and airborne users with a detailed scenario script.

The derived scenario retains 16 of the nodes, along with their names and location information. The nodes were chosen to produce a realistic example involving military equipment that can potentially be in a given region over a given time span. While the MITRE scenario has all the nodes working on a common mission, the derived scenario makes no such qualification.

The 16 nodes retained for Scenario 2 include 3 KC-135 tankers, 1 F-22, 1 E-3 AWACS, 1 RC-135V/W Rivet Joint, 1 E-8A Joint Surveillance Target Attack Radar System (Joint STARS), 1 A-10, 2 UASs, 1 Navy Battle Group (Navy_BG), 1 HH-60G Pave Hawk Rescue Helicopter (R-H), 1 Joint Strike Fighter (JSF), a Wing Operations Center (WOC), an Air & Space Operations Center (AOC), and a Joint Search and Rescue Center (JSRC). See Figure 20 for an overview of the assets and trajectories as represented

in OPNET Modeler 15.0. The octagons symbolize wireless nodes, and the white ovals indicate the orbits of the mobile nodes. The dimensions of the region shown measure roughly 550 km by 1,350 km.

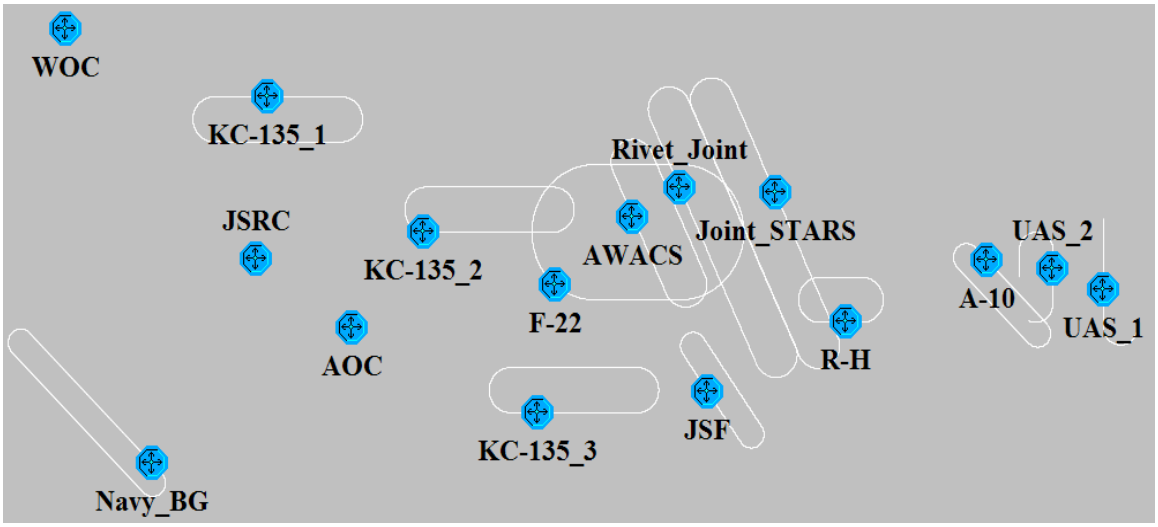

Figure 20: Scenario 2 overview

For Scenario 2, the R-H needs to send a constant stream of 1,024-bit packets at the rate of 1 packet/sec over a one-hour period to the JSRC. The JSRC is the only entity in the scenario that needs to receive this transmission, however the distance between the R-H and the JSRC precludes direct communication. Therefore, the messages must be passed through intermediate nodes, on possibly unrelated missions, in order to reach the JSRC.

As in Scenario 1, this scenario is split into two cases. The first case corresponds to the present situation, where no NTO process is in place to assist in *GIG-awareness*. The second case corresponds to a prospective situation, where an NTO process is in place. With the NTO process, advance trajectory information is available. The orbits of

pertinent participants can be known beforehand. However, the precise positions within those orbits are unknown until the aircraft are actually in the air.

For case 1, when no advance trajectory information is available, the R-H broadcasts its messages with intermediate nodes echoing the broadcast in order to reach the JSRC. Since only one traffic flow is modeled, no particular routing algorithms are employed. Each node keeps track of packets it has broadcast to avoid rebroadcasting packets multiple times. No carrier sensing is performed to avoid collisions or interference.

For case 2, when advance trajectory information is available through an NTO process, a specific route can be planned. In general, broadcasting a message meant for a single recipient is avoided because it is wasteful of bandwidth and may result in nodes receiving many copies of the same packet. In the planning pre-NTO phase, analysts determine which assets will be in the correct area when the transmissions need to be made. By plotting these assets and their planned orbits onto a map, a picture similar to Figure 20 can be generated. The goal is to have each hop bring a packet from the R-H closer to the JSRC. Each link needs to be of good quality and the total number of hops is minimized. Visual inspection of Figure 20 indicates that the A-10, WOC, Navy_BG, and two UASs are not good candidates for intermediate nodes. Using existing tools such as Spectrum XXI[9], analysts can consider terrain obstacles, distance, and enemy eavesdropping abilities to help plan a route. The characteristics of the radio transmitters and antennas on the various assets can be taken from the capabilities database and used to

[9] SPECTRUM XXI is a "Windows based, automated spectrum management tool that supports operational planning as well as near real-time management of radio frequency spectrum with emphasis on assigning compatible frequencies and performing spectrum engineering tasks" [64].

predict the received power at various locations in order to estimate the quality of transmission.

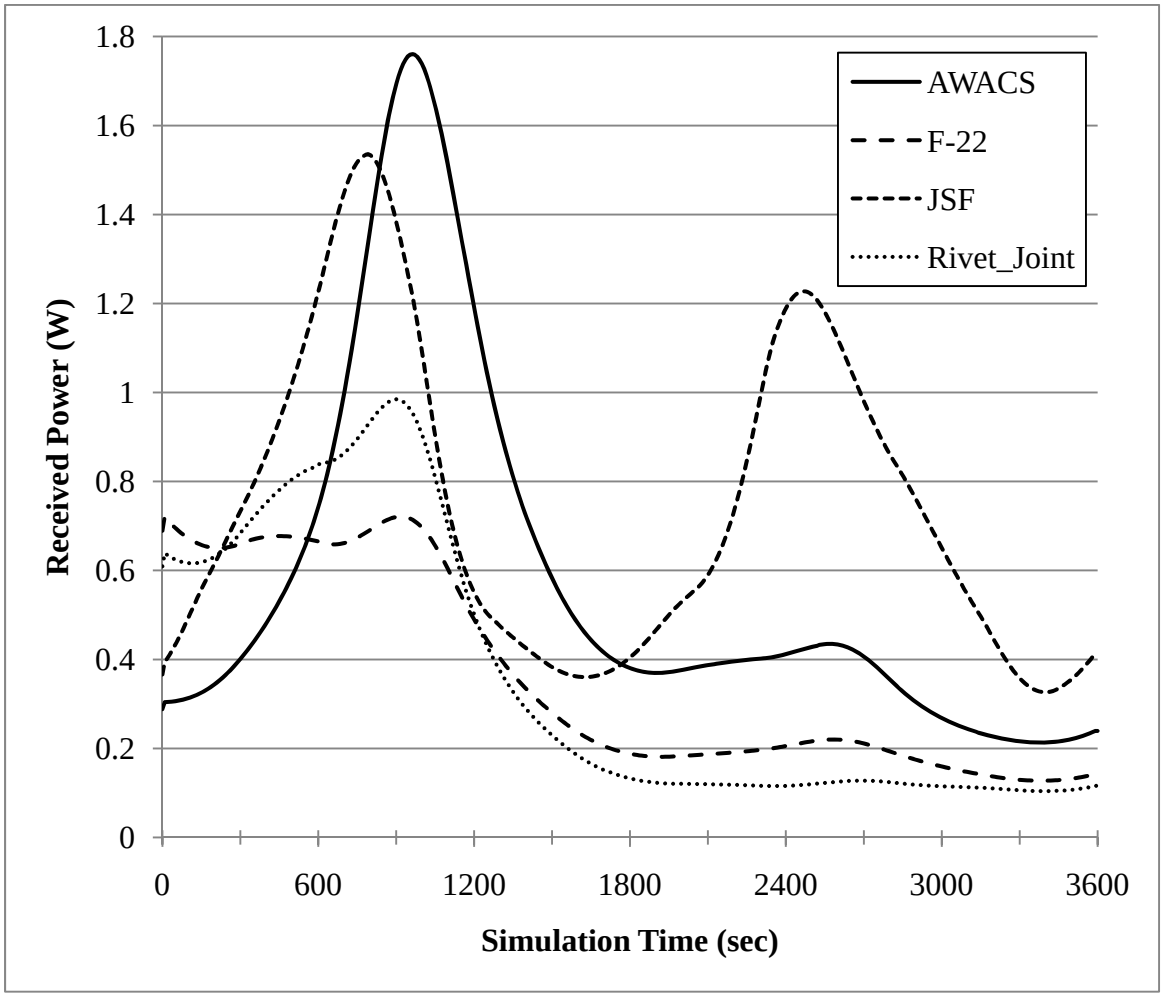

Figure 21: Example received power (W) from R-H in Scenario 2

Figure 21 above shows an example, for four aircraft close to the R-H, of power received from transmissions originating at the R-H. The received power at these four aircraft is sufficiently high for the duration of the transmission to provide good quality links. The fluctuation in power received over time is a function of the distance between source and destination as the aircraft navigate their respective orbits. These graphs show

only one pattern determined by the relative starting positions within the orbits of the various aircraft. A minimum power received level can be calculated based on the maximum expected distance between pairs of aircraft. Received power at KC-135_2 is not shown in Figure 21 because at the far ends of their orbits, R-H and KC-135_2 are not in range of each other. The received power at Joint_STARS is also not shown because of its orbit's similarity to the orbit of Rivet_Joint.

Using a methodology such as the above, analysts determine that the best route for packets is R-H → AWACS → KC-135_2 → KC-135_1 → JSRC. Accordingly, each of these aircraft is given a tasking in the NTO to implement this route. Flooding the network can be kept as a backup option in case one of the intermediate aircraft is unable to complete its mission.

Both cases of this scenario are executed in OPNET Modeler, version 15.0.A PL1. Every node in this scenario except for the JSRC and the R-H has the same general node model design. Each node is assigned[10] a minimum frequency as listed below in Table 11. The node model is designed to receive transmissions at one of these frequencies and to echo on either all other frequencies (for case 1) or on one specified frequency (for case 2). The node model for the JSRC only needs to receive transmissions, so no radio transmitters are included in its design. The node model for the R-H includes a simple source generator and no radio receiver. A more detailed description of the node models follows.

[10] In practice, frequency assignments are contained in the Joint Communications Electronics Operating Instruction (JCEOI) created by frequency managers in the JAOC.

Table 11: Frequency assignments for nodes in Scenario 2

Node name	Assigned Frequency (MHz)
A-10	300
AOC	320
AWACS	340
F-22	360
JSF	380
JSRC	400
Joint_STARS	420
KC-135_1	440
KC-135_2	460
KC-135_3	480
Navy_BG	500
R-H	520
Rivet_Joint	540
UAS_1	560
UAS_2	580
WOC	600

The node model shown below in Figure 22 is for KC-135_2, but represents the general design for all nodes except for the R-H and JSRC. Icons a_0 and a_1 represent isotropic antennas with a 50 dB gain in all directions. Icon rr_0 is a radio receiver with a data rate of 1,024 bps and a bandwidth of 10 kHz. The minimum frequency for rr_0 is set using the values in Table 11. The proc icon is a processor that examines each incoming packet. If the packet has been seen before, the packet is discarded. If the packet is new, it continues through to q_0. Icon q_0 represents a queue with a first in, first out service discipline. The queue acts as its own server with a service rate of 9,600 bps. For case 1, q_0 makes 14 copies each packet and sends the resulting 15 packets out to the radio transmitters. For case 2, q_0 simply sends each packet to the radio transmitter corresponding to the next hop on its route. The icons rt_0 through rt_15 are radio transmitters. Each transmitter is set to a different minimum frequency as listed in Table

11. Since it is not necessary for a node to transmit messages to itself, the node model omits the corresponding transmitter. For example, KC-135_2 is the ninth entry in Table 11, thus the ninth transmitter, rt_8, is not shown in Figure 22. Each transmitter has a data rate of 1,024 bps, a bandwidth of 10 kHz, and a transmission power of 1,000 Watts (W).

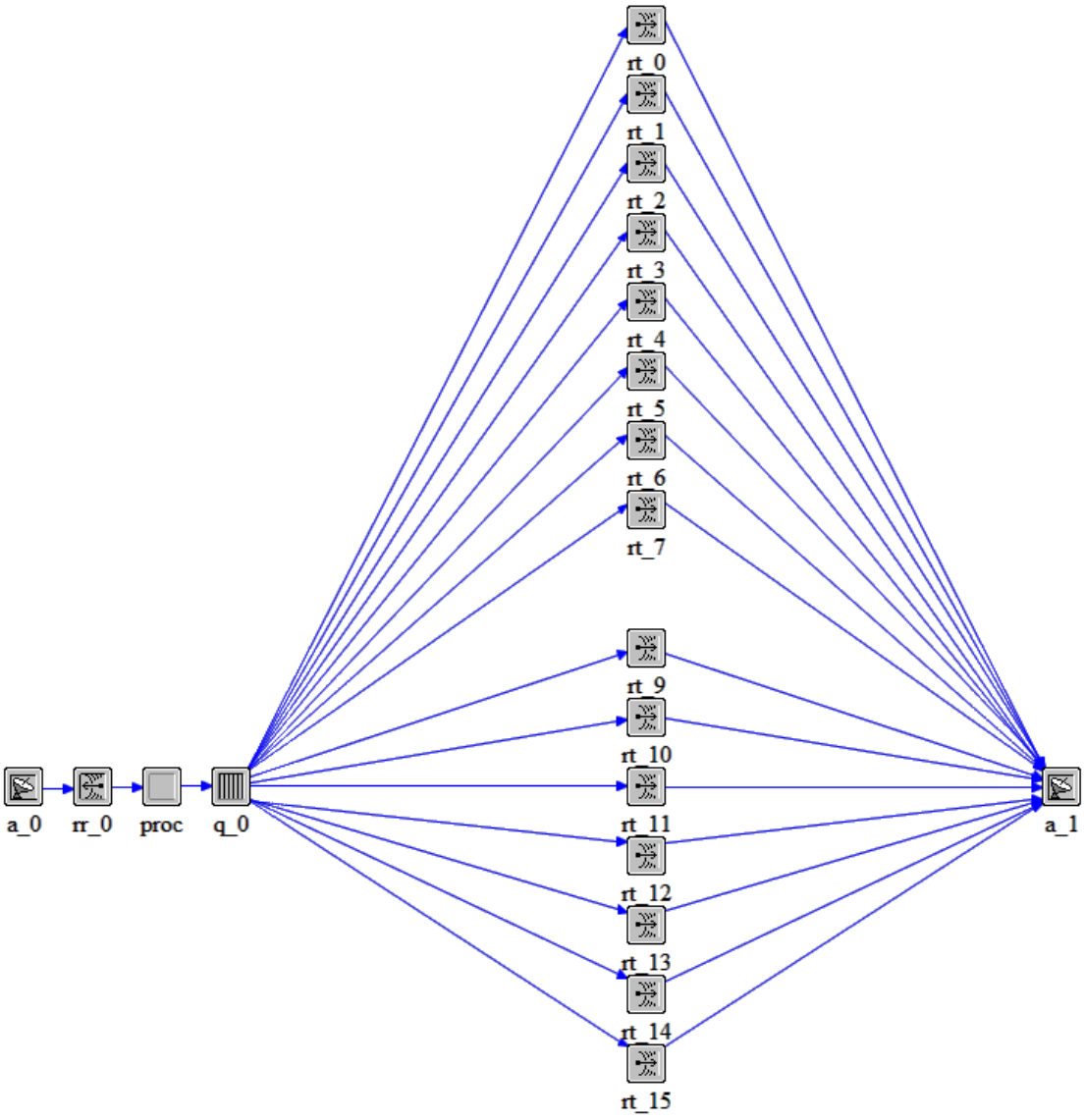

Figure 22: General node model for Scenario 2

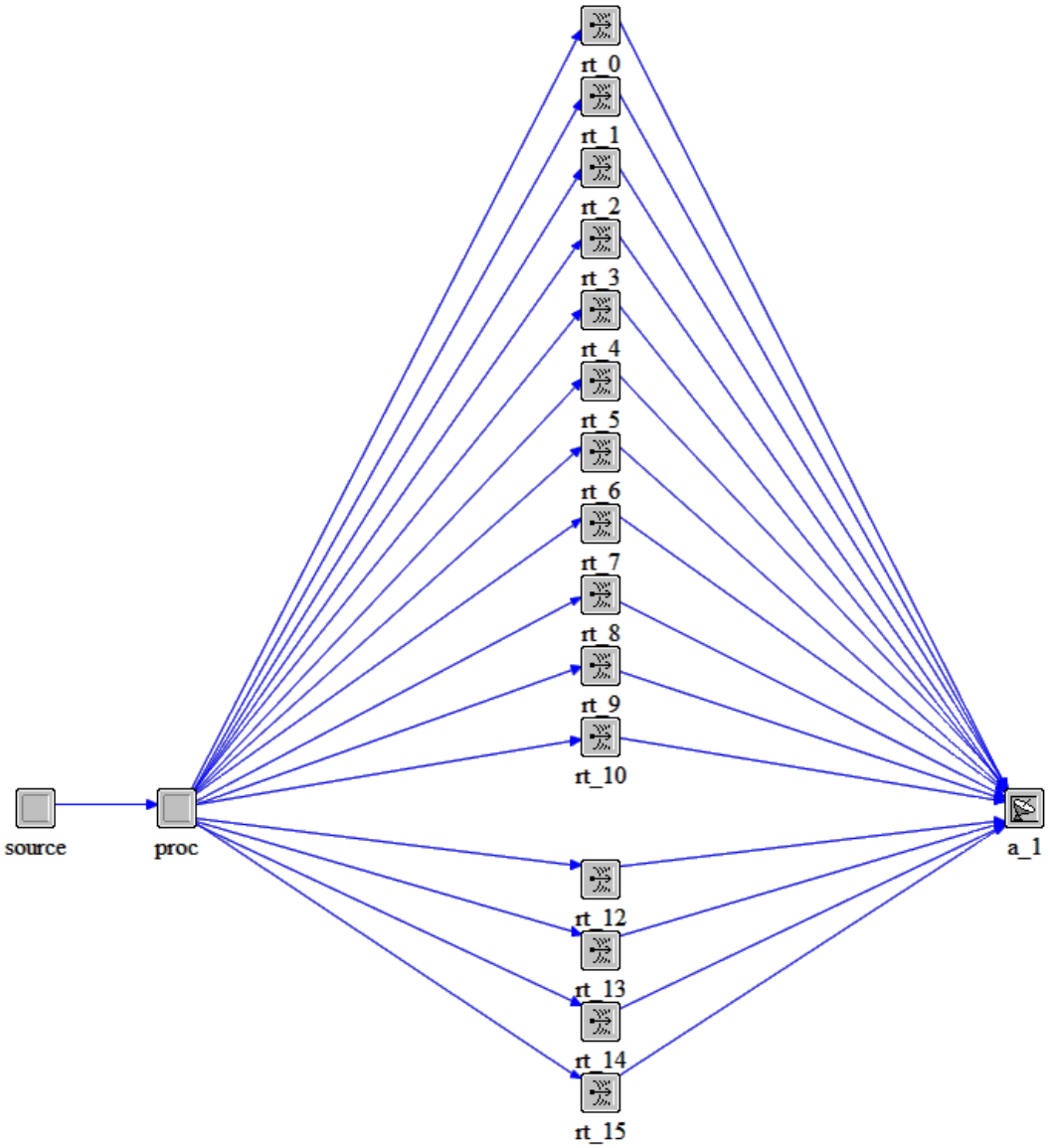

Figure 23: Node model for the R-H in Scenario 2

The node model for the R-H is shown above in Figure 23. The icons rt_0 through rt_15 and icon a_1 are defined exactly as in the general node model. The source icon represents a simple source processor that generates 1,024-bit packets with a constant interarrival time of 1 second. The source processor performs packet generation for a one-hour time span during the simulation. The proc icon is a processor that gives each packet

a unique sequence number. The sequence number allows any node that receives a packet to know whether or not the packet has been rebroadcast before or not. For case 1, proc makes 14 copies of each packet and sends the resulting 15 packets out to the radio transmitters. For case 2, proc simply sends each packet to rt_2, the radio transmitter set to the minimum frequency for the AWACS. Since the R-H is the source node, there is no need for its node model to have a radio receiver or receiving antenna.

The node model for the JSRC is shown below in Figure 24. The icons a_0, rr_0, proc, and q_0 are defined exactly as in the general node model. The sink icon represents a sink processor. Since the JSRC is the destination node, there is no need for its node model to have radio transmitters or transmitting antenna.

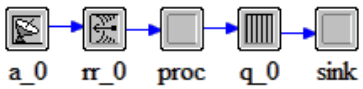

Figure 24: Node model for the JSRC in Scenario 2

The network metrics of end-to-end delay in seconds and traffic received in packets/sec are measured at the sink in the JSRC. Received power in W and throughput in packets/sec are measured at the radio receiver rr_0 in all models. The simulation is run for a duration of one hour, starting at 0 seconds and ending at 3,600 seconds. The random seed value is set to 100.

The results for Scenario 2 without an NTO process in place are examined first. In Figure 25, a graph of the received power (in W) for the radio receiver at the JSRC is given. The majority of the power received at the JSRC originates from the AOC, which, like the JSRC, is a fixed node. The fact that both the JSRC and the AOC are fixed

accounts for the horizontal portions of the graph. The changes in power level can be explained by interference during periods where aircraft such as the KC-135_1 and KC-135_2 are in range of the JSRC.

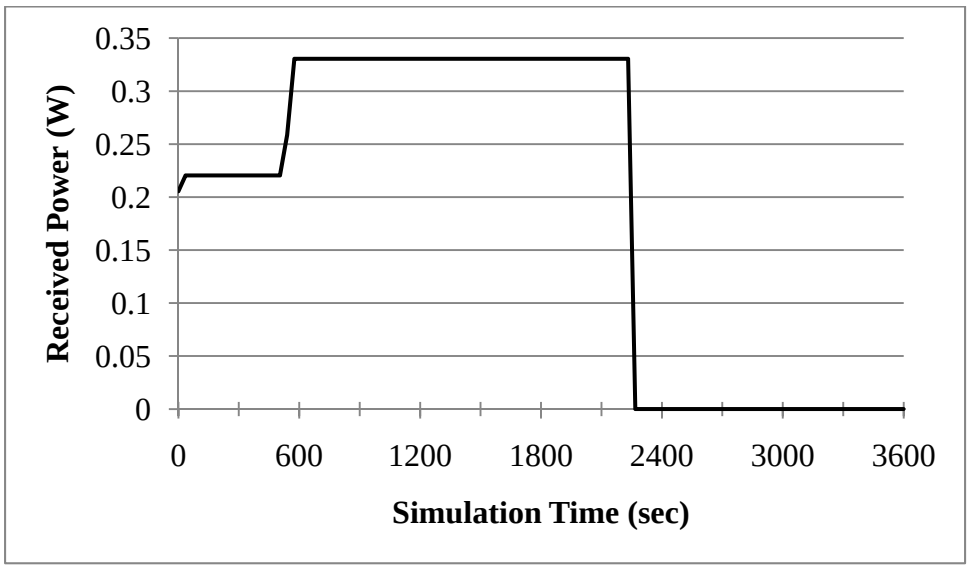

Figure 25: Received Power (W) at JSRC rr_0 in Scenario 2 (no NTO)

In Figure 26, the throughput in packets/sec for the radio receiver at the JSRC is shown. There is a short period around 2300 seconds and a longer period after 3200 seconds where interference does not allow any packets to be received. Even though the R-H is only producing packets at a rate of 1 packet/sec, the JSRC receives packets at a higher rate for the majority of the simulation. Since the network is being flooded in case 1, a packet may be received from multiple sources at the same time. Alternatively, an earlier packet that was routed through more hops may arrive at the same time as a later packet that went through a shorter route. In the node model for the JSRC, the radio receiver (icon rr_0) is placed before the processor (icon proc) that destroys packets that

have already been received. The next figure shows the throughput of unique packets at the sink processor of the JSRC.

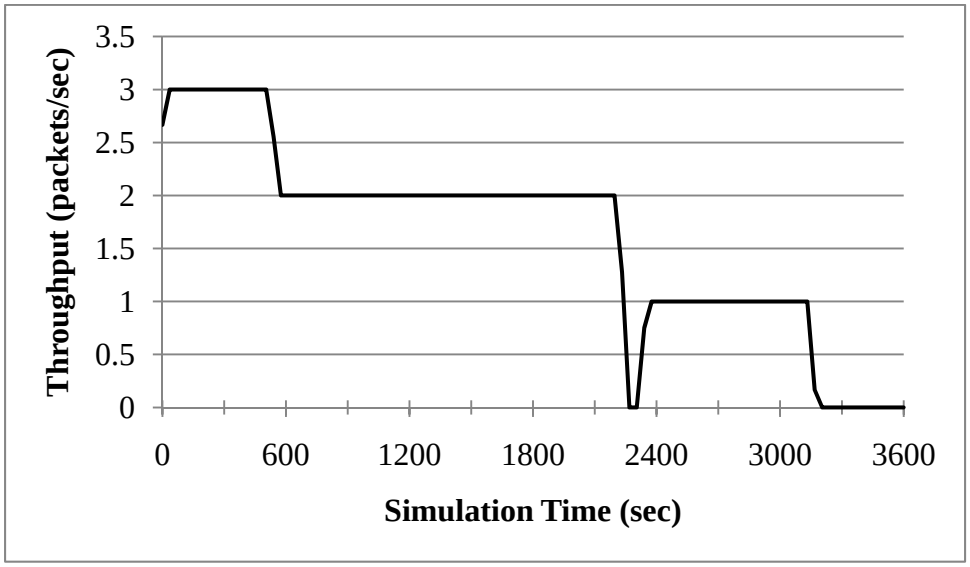

Figure 26: Throughput (packets/sec) at JSRC rr_0 in Scenario 2 (no NTO)

Once duplicate packets are destroyed, the surviving packets move on to the sink processor. Figure 27 below shows the rate in packets/sec at which traffic is received at the sink in the JSRC. The periods where no packets are received correspond to the same periods of zero throughput in Figure 26 above. However, the excess throughput at the radio receiver is reduced to the expected 1 packet/sec at the sink. Since there are some periods where no packets are received, it is relevant to find out what percentage of the packets sent from the R-H actually made it to the JSRC. In addition, it is interesting to determine how many packets were destroyed at the JSRC as well as at all of the other nodes in the network. These results are discussed next.

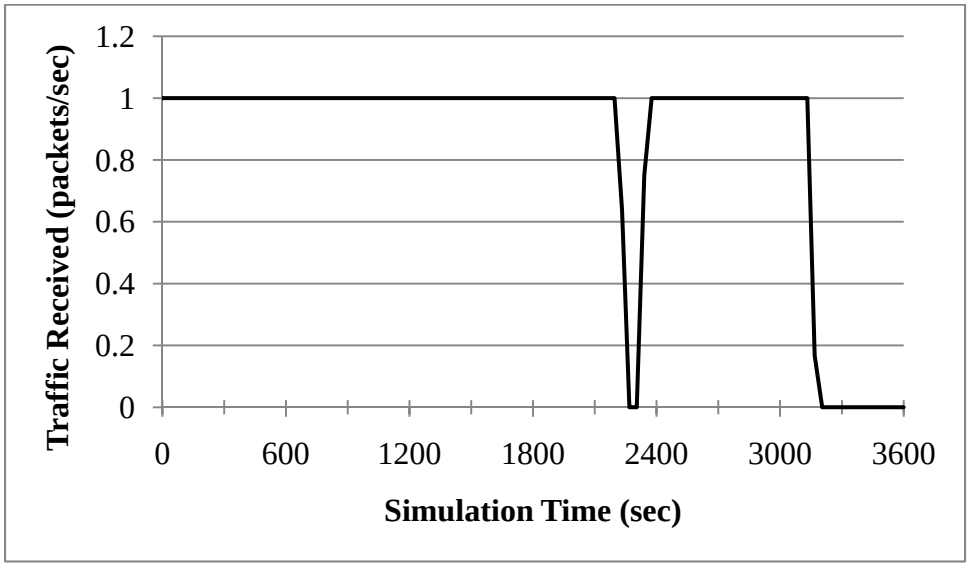

Figure 27: Traffic Received (packets/sec) at JSRC sink in Scenario 2 (no NTO)

Table 12 lists packet statistics for all of the nodes in Scenario 2. The first column lists the node names. The second column indicates how many duplicate packets were destroyed before being retransmitted. The third column lists how many unique packets were retransmitted. The fourth column catalogs the number of unique packets received as a percentage of the total number possible. As the source for the packets, the R-H creates 3,600 packets in a one-hour time frame. Without a radio receiver in its node model, the R-H neither receives nor destroys any packets. At the destination, the JSRC destroys 2,803 duplicate packets. The JSRC's node model does not include any radio transmitters, so the JSRC does not forward any packets. Of the 3,600 packets sent, 3,080 packets arrived at the final destination. That is, about 85.56% of the packets from the R-H were correctly received by the JSRC. Even though only the JSRC needed the information contained in the packets, every node received packets from the R-H except for the Navy_BG, which was out of range during the simulated time span. Hence, 15 out of 16

(or 93.75%) of the nodes in the scenario were involved. Of the nodes that received packets, only the WOC did not receive any duplicates. A total of 36,436 duplicate packets were destroyed at 13 nodes, averaging just over 2,800 duplicate packets per node. Thus, on average, these 13 nodes received about 77.85% more packets than required for a complete set. Perhaps the worst case is the KC-135_2 which only received 3,174 of the 3,600 unique packets (88.17%), but destroyed 4,498 duplicate packets. Those 4,498 packets destroyed plus the 3,174 packets that were forwarded equate to the KC-135_2 performing about 2.13 times the amount of work performed by the R-H. The Rivet_Joint's work performed is roughly 2.37 times the R-H's, but at least the Rivet_Joint was able to forward 100% of the original message.

Table 12: Packet statistics for nodes in Scenario 2 (no NTO)

Node	Packets Destroyed	Packets Forwarded	% Received
A-10	771	3600	100
AOC	2249	2249	62.47
AWACS	3623	3600	100
F-22	3678	3600	100
JSF	2249	3600	100
JSRC	2803	N/A	85.56
Joint_STARS	4808	3600	100
KC-135_1	1799	2249	62.47
KC-135_2	4498	3174	88.17
KC-135_3	3810	3600	100
Navy_BG	0	0	0
R-H	N/A	3600	N/A
Rivet_Joint	4919	3600	100
UAS_1	370	3600	100
UAS_2	859	3600	100
WOC	0	1799	49.97
[Total]	36436	41871	

In Figure 28, the end-to-end (ETE) delay in seconds is shown. The range of values is pretty small. The delay does not significantly change during the simulation, except for after 3,200 seconds, when the reception at the JSRC drops to zero. This graph only shows the ETE delay for those packets that made it all the way to the JSRC's sink processor. The destroyed duplicate packets and the absent packets are not included. The average ETE delay for the 3,080 packets that reached the sink is approximately 0.2140 seconds.

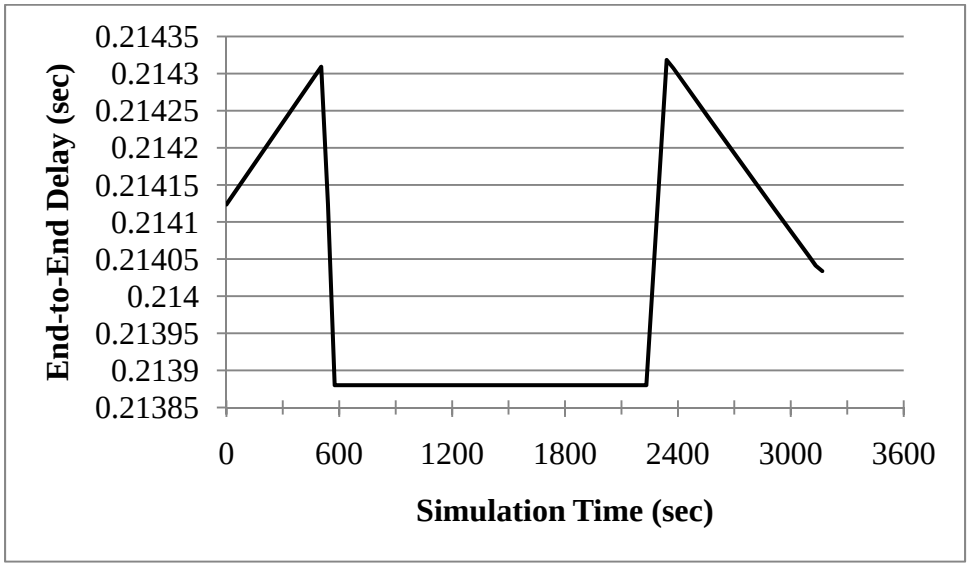

Figure 28: ETE Delay (sec) at JSRC sink in Scenario 2 (no NTO)

The results for Scenario 2 with an NTO process in place are examined next. In Figure 29, a graph of the received power (in W) for the radio receiver at the JSRC is given. Since the JSRC is receiving transmissions from the AOC, and both are fixed nodes, the power received at the JSRC remains constant. The received power for case 2 is about double the maximum received power seen in case 1. The difference is explained by how OPNET calculates received power. Validation experiments indicate that OPNET

averages the received power from each source to obtain a combined received power at a given radio receiver.

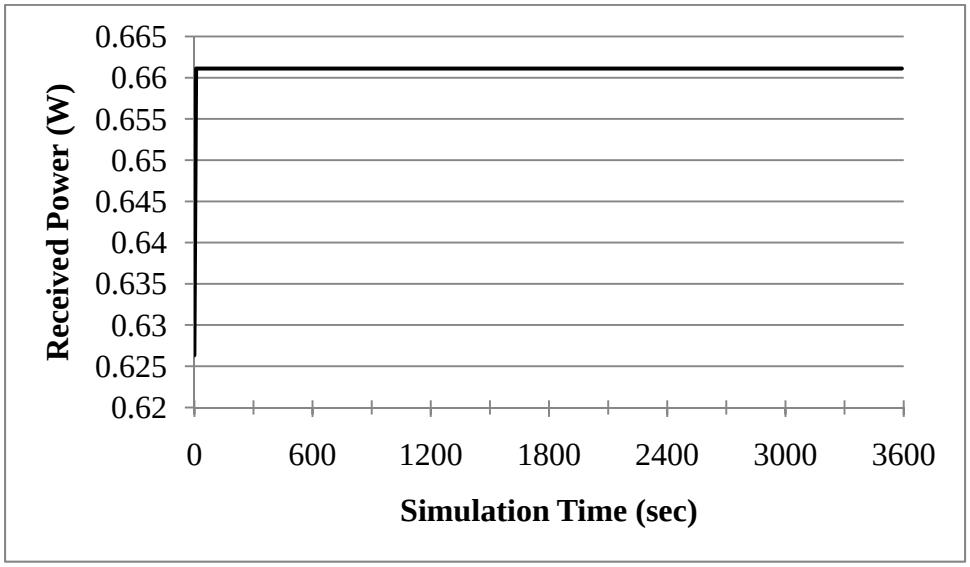

Figure 29: Received Power (W) at JSRC rr_0 in Scenario 2 (with NTO)

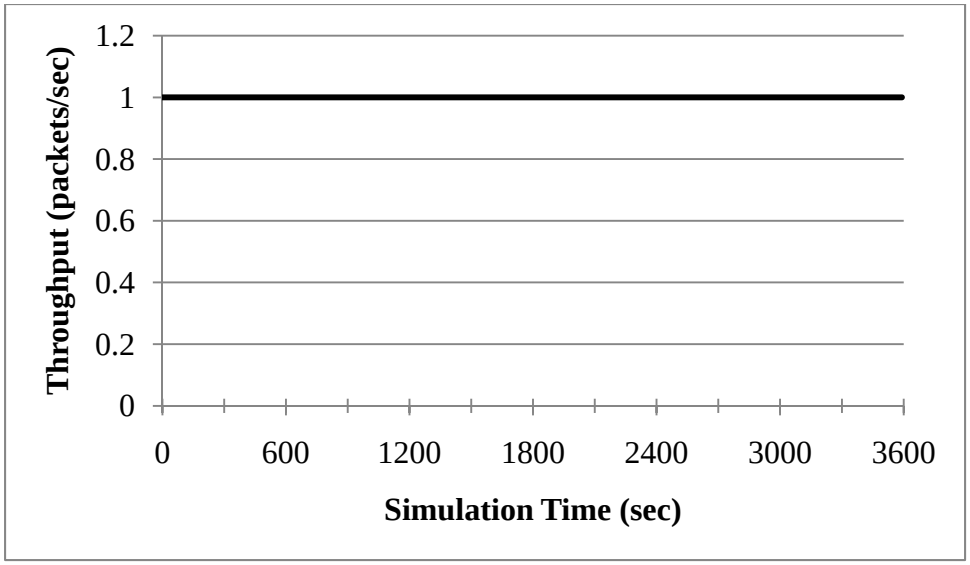

Figure 30: Throughput (packets/sec) at JSRC rr_0 in Scenario 2 (with NTO)

The throughput in packets/sec at the radio receiver for the JSRC is shown above in Figure 30. Since each hop on the path from the R-H to the JSRC is only receiving

packets from the previous hop's transmitter, the throughput does not exceed the speed of packet creation. Some fluctuations might have appeared had any packets been lost on the route or had traffic backed up at a queue. However, the data rates for the queues and radio equipment was sufficiently high to avoid any such problem.

Because packets follow a specified route, no duplicate packets are ever created. Thus the throughput at the radio receiver matches the traffic received at the sink for the JSRC. All packets received at the receiver are sent along to the sink. Thus, the graph for traffic received in Figure 31 is virtually identical to the graph in Figure 30.

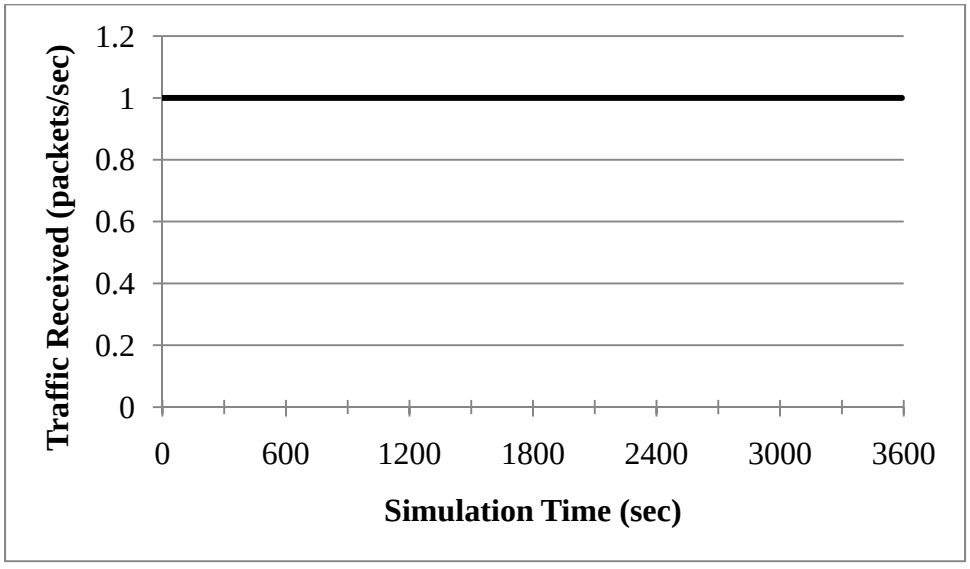

Figure 31: Traffic Received (packets/sec) at JSRC sink in Scenario 2 (with NTO)

Table 13 lists packet statistics for all of the nodes in Scenario 2 with routing (case 1 – using NTO) rather than flooding (case 2 – no NTO). The columns are as defined for Table 12. The source at the R-H created 3,600 packets in a one-hour time frame and sent them only to the AWACS. The AWACS received 100% of the packets and forwarded them all to the KC-135_2. The KC-135_2 received 100% of the packets and sent them on

to the AOC. The AOC also received 100% of the packets and passed them along to the JSRC. Finally, the JSRC received 100% of the packets and no further routing was performed. Out of 16 nodes in the scenario, only 5 (or 31.25%) were involved in the data transfer. No duplicate packets were destroyed anywhere, because no duplicate packets were created and each packet was sent only once by each node on the path.

Table 13: Packet statistics for nodes in Scenario 2 (with NTO)

Node	Packets Destroyed	Packets Forwarded	% Received
A-10	0	0	0
AOC	0	3600	100
AWACS	0	3600	100
F-22	0	0	0
JSF	0	0	0
JSRC	0	N/A	100
Joint_STARS	0	0	0
KC-135_1	0	0	0
KC-135_2	0	3600	100
KC-135_3	0	0	0
Navy_BG	0	0	0
R-H	N/A	3600	N/A
Rivet_Joint	0	0	0
UAS_1	0	0	0
UAS_2	0	0	0
WOC	0	0	0
[Total]	0	14400	

Finally, the graph in Figure 32 shows the ETE delay in seconds for packets created at the R-H and reaching the sink at the JSRC. The fluctuations are due to the change in route length as the AWACS and KC-135_2 traverse their orbits. The average ETE delay is about 0.8560 seconds. The average ETE delay for case 2 is about four times longer than the ETE delay for case 1. The extra delay is expected because flooding the

network results in packets taking multiple paths to the destination. Naturally, packets taking the shortest of the paths arrive at the sink first, and those are the packets whose delay is recorded. Unless the specified route is also the shortest route, flooding has shorter delays.

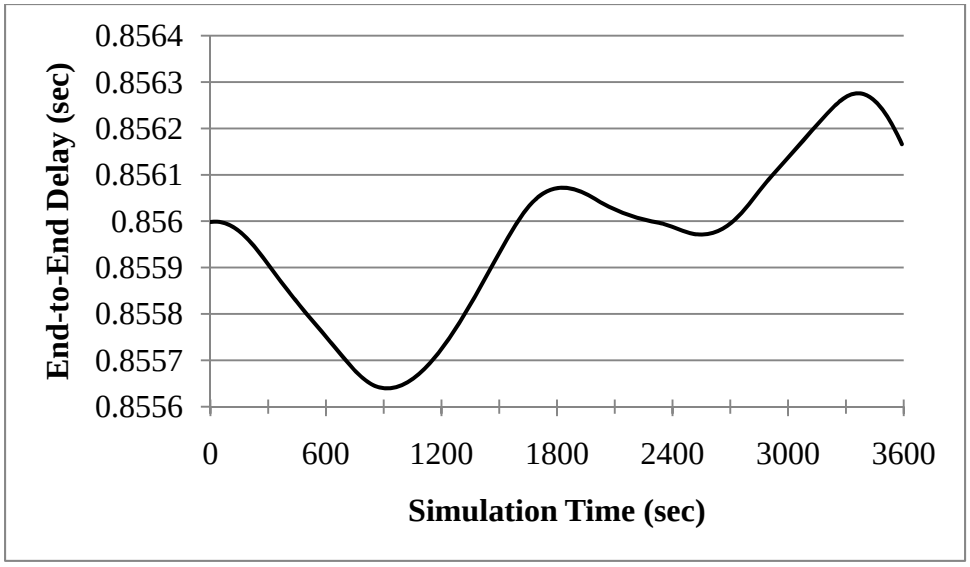

Figure 32: ETE Delay (sec) at JSRC sink in Scenario 2 (with NTO)

In terms of packet statistics, having an NTO for Scenario 2 presents a marked improvement for several reasons. The message from the R-H is only needed by the JSRC. With flooding, 93.75% of the nodes in the area were involved in the transfer as opposed to 31.25% with routing. With flooding, some nodes were performing the equivalent of sending more than two full messages, but actually handling less than 100% of the message. With routing, nodes performed exactly enough work to send 100% of the message. With flooding, the JSRC only received 85.56% of the message because of interference while routing the message was lossless.

In terms of ETE delay, having an NTO for Scenario 2 seems, at first glance, to be detrimental. Had the scenario involved interactive audio/video or remote instrument control, shorter delays would be necessary. However, the data loss associated with case 1 is unacceptable for such applications. For non-interactive teledata purposes, such as modeled here, application quality can remain uncompromised even with latencies on the order of seconds [62:8]. The peak delay for case 2 is less than one second, which is still tolerable for this particular scenario.

Some improvements can be made to case 1 of Scenario 2 by making the flooding behavior more sophisticated. For example, nodes can be instructed to mark each packet they forward with an identifier. In this way, any node that receives a packet can refrain from forwarding that packet back to the node from which it was received. This results in fewer destroyed packets at each node and less wasted bandwidth. Another improvement can be made by implementing carrier sensing to alleviate interference and collisions.

4.1.3 NTO Scenario 3

In [54, 55], Göçmen illustrates the benefits of an NTO in a CSAR execution phase scenario using the JTRS software-defined radio system through two experimental simulations. For his scenario, an injured pilot in a deep valley sends a beacon signal that is located by satellite and relayed to headquarters. A CSAR Task Force (CSARTF) with recovery vehicle is already en route to the pilot's last known location. Headquarters needs to send the exact location of the pilot as a critical image update to the CSARTF. As in the previous two scenarios, this scenario is broken down into two cases. In the first case, the data sent by headquarters is broadcast to the recovery vehicle through every available

asset. In the second case, data is flooded along four different routes designated by an NTO. Each of these two cases is examined under high and low traffic loads.

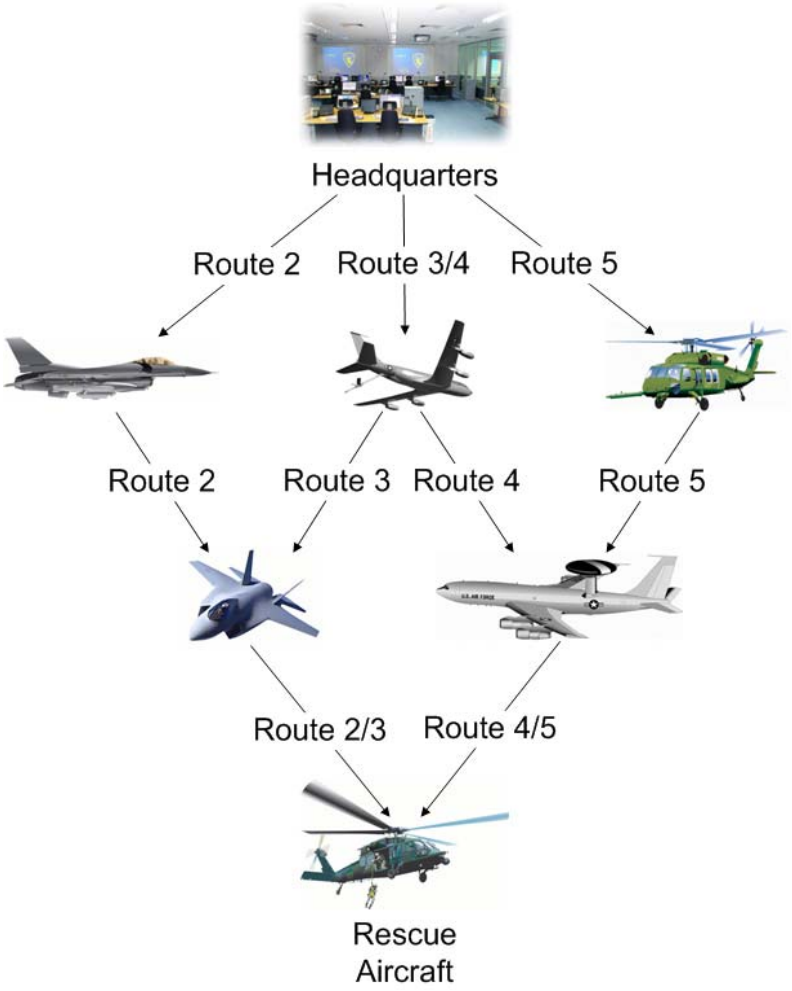

Figure 33: Scenario 3 overview

Göçmen's CSAR scenario was implemented using OPNET Modeler 14.0. A total of six aircraft are employed in the 45-minute scenario. When traffic is flooded in the case without an NTO process, the flooding is referred to as Route 1. For the case where an NTO process specifies a route, the four routes are numbered 2 through 5. See Figure 33 above for an overview of the assets and routes involved.

Even though the assets shown are all of different types, they are assumed to have a common JTRS. Transmission speed was set at 90 Kbps using 5,000-bit packets with a constant interarrival time. For high traffic loads, the interarrival time was set at 0.0625 seconds. For low traffic loads, the interarrival time was set at 0.1667 seconds. The metric measured for all cases and traffic loads was the end-to-end delay. Each experiment was run ten times with different random seeds. Since there are five different routes subjected to two traffic loads, a total of 100 simulation runs were performed.

Göçmen hypothesized the end-to-end delay to be longer when packets are flooded than when a specific route is utilized regardless of traffic load. He found his hypothesis to be true under high traffic loads, but that flooding was faster under low traffic loads. The results in this section come from Göçmen's thesis [55]. They are included here because his experiments were inspired by the results from Scenario 1 and were completed prior to the publication of this dissertation.

Figure 34 is a scatter plot showing the ETE delay in seconds for all five routes under a high traffic load. The figure shows Route 1 (flooding) experiencing the longest ETE delays for the majority of the simulation time. The plots for Route 2 and Route 3 can be seen to overlap, as do the plots for Route 4 and Route 5. The overlap can be explained by the routes having common last hops. As the pairs of routes intersect at the last hop, packets begin to queue up under high traffic loads. Packets from each source experience similar delays as they wait to be forwarded.

Figure 35 shows the 95% confidence intervals for mean ETE delay in seconds using the results from ten random seeds. The confidence intervals support Göçmen's

hypothesis that ETE delay is shorter for the NTO designated routes under a high traffic load. Note that the interval for Route 1 (flooding) does not overlap the intervals for any of the specified routes.

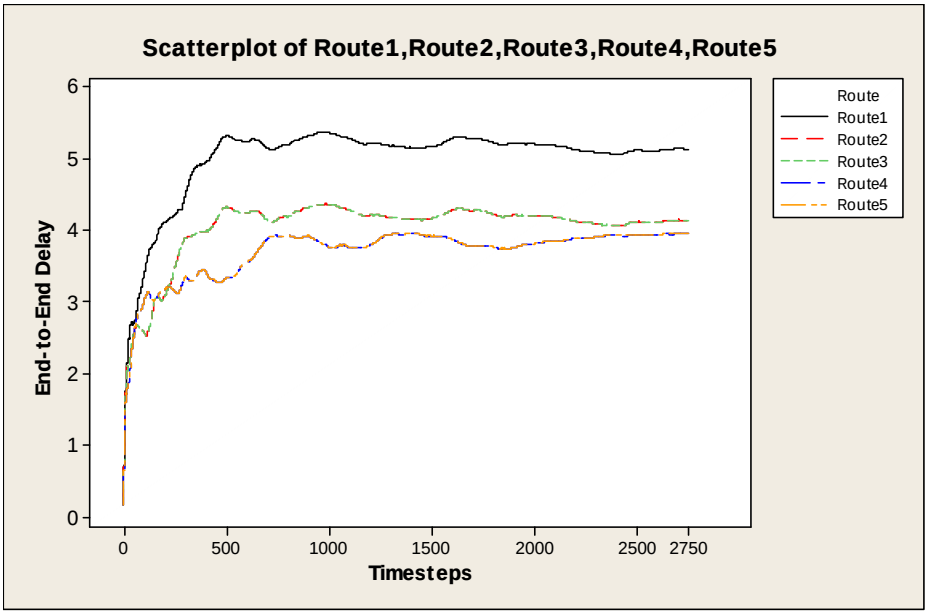

Figure 34: ETE delay (sec) for Scenario 3 (high traffic load) [55:36]

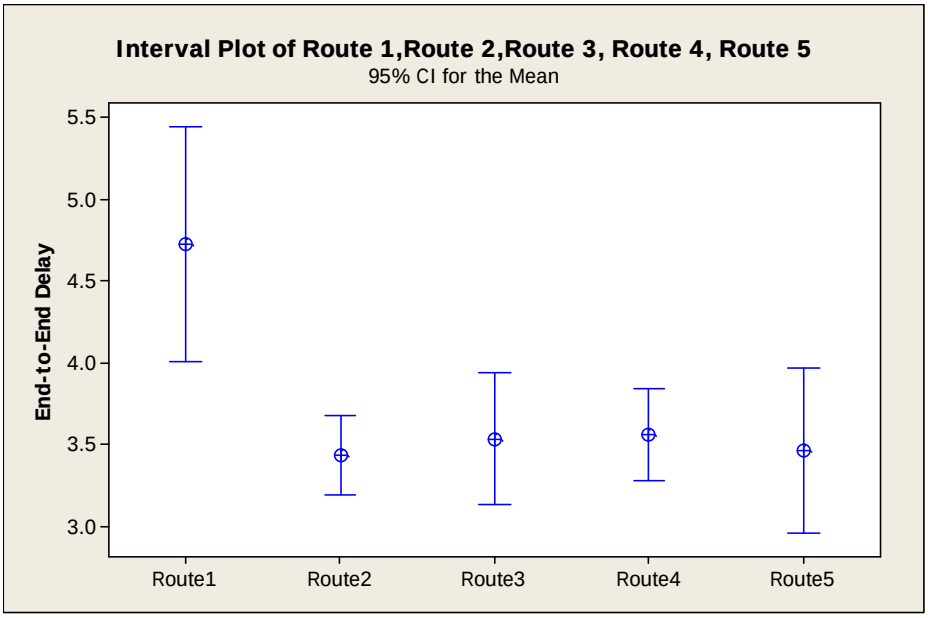

Figure 35: 95% CI for mean ETE delay (sec) for Scenario 3 (high traffic load) [55:37]

Figure 36 is a scatter plot showing the ETE delay in seconds for all five routes under a low traffic load. The figure shows Route 1 (flooding) experiencing the shortest ETE delays for the majority of the simulation time.

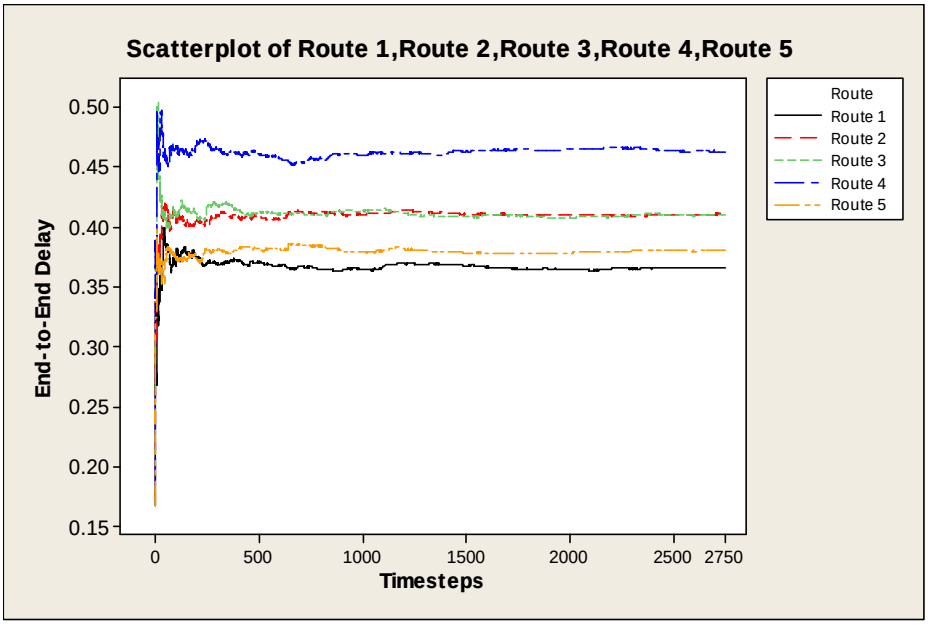

Figure 36: ETE delay (sec) for Scenario 3 (low traffic load) [55:39]

Figure 37 shows the 95% confidence intervals for mean ETE delay in seconds using the results from ten random seeds. Here, the confidence intervals refute Göçmen's hypothesis that ETE delay is shorter for the NTO designated routes under a low traffic load. There is a small overlap between the intervals for Route 1 and Route 5, but clearly, the NTO-designated routes face higher latency.

As in Scenario 2, this scenario shows how the existence of an NTO process can improve the QoS of the GIG. Under both high and low traffic loads, bandwidth is saved. Under high traffic loads, ETE is less (~1 sec) when an NTO-designated route is used. Under low traffic loads, ETE is greater (~0.1 sec) with an NTO-designated route. One

can argue, however, that the GIG in a CSAR mission is not likely to be lightly loaded. Also, the slight increase in ETE in a low load situation is justified by a greater improvement in overall bandwidth availability and potential decrease in interference.

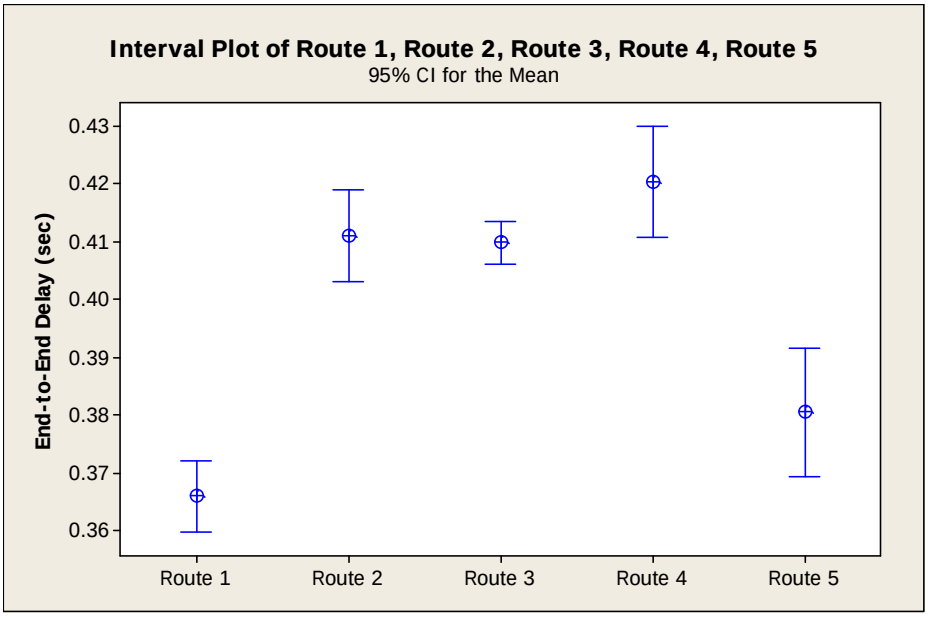

Figure 37: 95% CI for mean ETE delay (sec) for Scenario 3 (low traffic load) [55:40]

4.2 Polymorphic Networking Problem

In this section the metrics of solution time, true solution cost, Δ distance, network diameter, and average number of hops for commodities are presented for the various network configurations outlined in Chapter Three (III). All configurations for networks with 5 to 20 nodes are solved for 10 polymorphisms with 30 randomly generated input files. This gives 300 samples for each configuration with which to generate confidence intervals for results. These results are given in the first subsection. Because of the nature of the MILP formulation of the PNP, it takes a very long time to generate a single polymorphism for large networks, much less 10 polymorphisms. As a result, a full set of

30 randomly generated test cases are not run for all networks with 25 to 40 nodes. In some configurations, only a single test case was able to be run to completion. Six of the larger 40-node configurations and one of the 35-node configurations were not performed. The results for these configurations are offered in the second subsection. The confidence intervals there are larger in general, or nonexistent.

4.2.1 Networks of 5-20 Nodes

All configurations for networks with 5 to 20 nodes have been solved for 10 polymorphisms with 30 randomly generated input files. The configurations are referenced by the shorthand '#N#C#I' where each '#' symbol is replaced by the actual number of nodes, commodities per node, and interface types. For example, 5N2C3I denotes a network configuration consisting of 5 nodes, where each node is the source of 2 commodities and has 3 interface types available. Each input file randomly assigns the destination nodes for commodities and the potential-adjacency matrix. Thus, for each configuration, there are 30 test cases (input files) that GAMS/CoinCbc runs through 10 iterations for a total of 300 topologies generated per configuration.

Tables 14-17 show the 95% confidence intervals for the mean and median times in seconds required to generate each polymorphism for each configuration. The minimum and maximum solution times out of the 300 topologies for each configuration are also given. The tables are designed for ease of comparison. Each table shows the results for a set number of nodes. To compare configurations that vary only in number of nodes, look at corresponding rows from each table. To compare configurations varying only in the number of commodities per node, look at commonly shaded rows within a table. For

configurations only differing by interface types per node, consider a group of four consecutive rows shaded from white to dark grey.

Table 14: PNP time results for 5-node configurations

Config.	95% Confidence Interval for Mean Time in Seconds		95% Confidence Interval for Median Time in Seconds		Min. Time in Seconds	Max. Time in Seconds
5N1C1I	0.31288	0.39826	0.16322	0.26400	0.059	2.001
5N1C2I	0.16817	0.20541	0.11755	0.12745	0.068	1.497
5N1C3I	0.40525	0.50076	0.26055	0.33100	0.080	2.289
5N1C4I	0.33866	0.46356	0.16400	0.18100	0.084	2.897
5N2C1I	0.20297	0.26780	0.10000	0.11200	0.070	2.234
5N2C2I	0.12472	0.14379	0.10000	0.11400	0.082	1.051
5N2C3I	0.23106	0.31110	0.13255	0.17500	0.095	3.531
5N2C4I	0.17575	0.22977	0.14255	0.18934	0.111	2.813
5N3C1I	0.18283	0.21487	0.14400	0.15400	0.113	1.493
5N3C2I	0.20042	0.21757	0.18355	0.18845	0.169	1.336
5N3C3I	0.28377	0.30664	0.24800	0.26689	0.207	0.926
5N3C4I	0.34334	0.37033	0.29500	0.32345	0.251	1.130

Table 15: PNP time results for 10-node configurations

Config.	95% Confidence Interval for Mean Time in Seconds		95% Confidence Interval for Median Time in Seconds		Min. Time in Seconds	Max. Time in Seconds
10N1C1I	0.17837	0.20313	0.15655	0.16500	0.101	0.786
10N1C2I	1.06170	1.35600	0.35060	0.40560	0.157	4.297
10N1C3I	1.38000	1.54730	1.41360	1.59760	0.201	4.132
10N1C4I	2.74590	3.11330	2.59370	2.98370	0.293	7.798
10N2C1I	0.27998	0.31513	0.25955	0.27500	0.208	2.212
10N2C2I	0.78220	0.88212	0.65100	0.72745	0.406	2.595
10N2C3I	1.62410	1.84930	1.32860	1.52690	0.563	6.052
10N2C4I	3.00970	3.43020	2.54290	3.05310	0.718	12.675
10N3C1I	0.36812	0.40221	0.33800	0.35100	0.279	1.658
10N3C2I	1.03580	1.15060	0.90810	0.97830	0.552	3.065
10N3C3I	2.45870	2.80790	2.08410	2.42700	0.813	8.871
10N3C4I	4.22980	4.86960	3.68130	4.23590	1.056	24.586

Table 16: PNP time results for 15-node configurations

Config.	95% Confidence Interval for Mean Time in Seconds		95% Confidence Interval for Median Time in Seconds		Min. Time in Seconds	Max. Time in Seconds
15N1C1I	0.34589	0.36879	0.32511	0.35200	0.268	1.209
15N1C2I	2.91340	3.22170	2.83270	3.16810	0.536	8.715
15N1C3I	11.3070	13.5250	8.64600	10.5130	1.099	58.960
15N1C4I	37.3560	48.0260	23.0420	29.3320	1.862	328.274
15N2C1I	0.55224	0.57662	0.51755	0.53700	0.448	1.120
15N2C2I	3.42250	3.88170	3.02130	3.56020	0.976	14.255
15N2C3I	12.2080	14.9900	9.86400	11.5850	1.455	120.301
15N2C4I	51.8210	65.3710	37.7590	47.3210	2.653	503.496
15N3C1I	0.89848	0.98623	0.77111	0.79245	0.668	3.365
15N3C2I	4.62430	5.17570	4.11250	4.98180	1.531	13.702
15N3C3I	19.7440	25.7890	14.1600	16.7470	2.454	269.669
15N3C4I	109.960	141.330	72.4700	90.3300	4.042	889.007

Table 17: PNP time results for 20-node configurations

Config.	95% Confidence Interval for Mean Time in Seconds		95% Confidence Interval for Median Time in Seconds		Min. Time in Seconds	Max. Time in Seconds
20N1C1I	0.54250	0.55993	0.53300	0.54689	0.458	1.177
20N1C2I	6.40600	7.99800	5.47700	6.24500	1.052	103.283
20N1C3I	78.0300	115.430	39.9600	54.1400	2.941	1501.457
20N1C4I	685.500	1044.20	312.900	442.000	6.246	18238.805
20N2C1I	1.05160	1.09740	0.99560	1.02230	0.871	2.212
20N2C2I	7.75630	9.00090	6.92620	8.09030	2.048	44.131
20N2C3I	104.830	150.700	45.9400	71.1300	3.073	1599.392
20N2C4I	900.000	1364.60	351.100	511.700	23.504	15435.227
20N3C1I	1.74660	2.01750	1.42900	1.46540	1.275	11.911
20N3C2I	11.5390	13.4610	9.58700	11.6090	3.059	78.755
20N3C3I	121.800	166.390	58.7000	85.9400	4.636	1890.646
20N3C4I	1786.50	2812.70	628.700	962.300	19.981	37831.996

From these tables, it can be seen that solution time increases more dramatically with respect to the number of nodes and the number of interface types per node than with

respect to the number of commodities per node. Of these configurations, the longest solution time for any single iteration was about 10.5 hours for one case of configuration 20N3C4I.

To examine true solution costs across polymorphisms, the percentage change from optimum is considered. Because the MILP is complete and optimal, the first solution without domain modifications is an optimum solution. The percentage change from the cheapest solution to the most expensive polymorphism is calculated by taking the difference in true cost and dividing by the optimal value. For each configuration, 30 calculations are performed (one for each test case). Those 30 percentage changes are averaged. Tables 18-21 below show the results. The first column in the tables is the configuration shorthand, the second column contains the mean percentage change in cost, and the standard deviation is in the third column. In addition, the minimum and maximum percentage changes in cost are reported in columns four and five.

Table 18: True cost results for 5-node configurations

Config.	Mean % Change in Cost	Standard Deviation	Min. % Change in Cost	Max. % Change in Cost
5N1C1I	27.70	11.24	11.76	53.85
5N1C2I	33.45	14.83	12.50	64.29
5N1C3I	29.33	13.01	6.67	63.64
5N1C4I	32.13	12.24	13.33	63.64
5N2C1I	22.83	12.08	11.54	73.91
5N2C2I	24.66	8.14	13.04	52.17
5N2C3I	20.83	7.83	8.33	42.86
5N2C4I	21.76	7.89	8.70	40.00
5N3C1I	38.75	10.80	21.21	58.06
5N3C2I	25.38	7.20	12.90	45.16
5N3C3I	21.02	6.56	12.90	41.94
5N3C4I	21.05	4.89	9.68	30.00

Table 19: True cost results for 10-node configurations

Config.	Mean % Change in Cost	Standard Deviation	Min. % Change in Cost	Max. % Change in Cost
10N1C1I	28.93	12.70	4.08	56.76
10N1C2I	32.50	10.78	10.82	54.84
10N1C3I	34.39	10.94	19.44	68.00
10N1C4I	33.27	8.53	18.18	58.62
10N2C1I	29.99	11.69	14.75	55.00
10N2C2I	33.31	6.06	18.97	44.64
10N2C3I	29.45	6.96	18.18	49.06
10N2C4I	26.72	5.59	16.67	42.31
10N3C1I	31.92	8.35	17.02	50.59
10N3C2I	35.42	7.48	20.00	58.90
10N3C3I	33.33	5.44	24.32	45.07
10N3C4I	27.59	6.22	16.44	37.68

Table 20: True cost results for 15-node configurations

Config.	Mean % Change in Cost	Standard Deviation	Min. % Change in Cost	Max. % Change in Cost
15N1C1I	24.13	10.64	10.00	55.84
15N1C2I	31.50	7.34	16.95	48.28
15N1C3I	31.53	5.21	21.43	44.23
15N1C4I	31.44	5.41	22.22	45.28
15N2C1I	26.03	9.19	13.39	49.12
15N2C2I	33.59	5.55	25.00	43.33
15N2C3I	29.31	4.32	20.65	39.77
15N2C4I	27.85	4.58	19.77	37.21
15N3C1I	28.47	10.08	10.85	53.37
15N3C2I	36.69	6.71	26.52	54.40
15N3C3I	31.74	5.11	22.40	42.61
15N3C4I	29.10	3.86	23.08	39.50

Table 21: True cost results for 20-node configurations

Config.	Mean % Change in Cost	Standard Deviation	Min. % Change in Cost	Max. % Change in Cost
20N1C1I	20.61	13.28	4.38	61.61
20N1C2I	29.80	5.34	21.84	44.71
20N1C3I	28.61	5.64	22.50	45.21
20N1C4I	29.91	6.20	20.51	44.44
20N2C1I	19.79	8.19	2.90	40.10
20N2C2I	36.58	5.09	29.45	50.69
20N2C3I	31.83	4.41	22.73	42.52
20N2C4I	28.42	4.60	18.46	39.17
20N3C1I	19.63	5.52	6.71	29.61
20N3C2I	38.91	4.44	27.92	47.03
20N3C3I	34.12	3.76	26.37	41.04
20N3C4I	30.32	3.87	21.84	38.92

There does not appear to be any clear pattern in solution cost correlating to the number of nodes, the number of commodities per node, or the number of interfaces per node. The minimum and maximum percentage changes in cost do offer some interesting interpretations.

When the minimum percentage change is small, it implies that all of the polymorphisms generated for that case were very close to optimal. For example, consider the minimum percentage change of 4.08% for one case of configuration 10N1C1I. The sequence of polymorphism costs for this case is 49, 49, 50, 50, 50, 50, 51, 51, 51, and 51. The first two solutions are optimal and different from each other (with a Δ of 0.1111). The next four solutions consist of two different topologies that alternate. The last four solutions are all different from each other.

The largest maximum percentage change of 73.91% occurred for one case of configuration 5N2C1I. The sequence of polymorphism costs for this case is 23, 25, 27, 25, 28, 27, 23, 40, 23, and 24. Most of the polymorphisms stay within 21.74% of optimal. Typically, a commodity may have a few routes from source to destination involving one or two hops to choose from. From iteration to iteration, the additional cost to reuse those paths increases. Eventually, it becomes cheaper for the commodity to use a longer route with three or four hops involving edges that were not previously traversed. In this particular case, several commodities were forced to take longer routes at the same time, leading to a big jump in true cost. However, for the next iteration, the additional cost of those longer routes increased making the shorter routes more cost effective again. This case seems to be exceptional. It appears that the MILP approach to the PNP produces polymorphisms within 75% of optimum for the potential-adjacency matrices used here.

This bound of 75% is most certainly tied to the network topology described by the potential-adjacency matrix in the randomly generated input files. One can design a network in a ring topology in such a way that each node has a commodity that is destined one node clockwise in the network. Most polymorphisms for such a network utilize one-hop routes for the commodities. However, as the edges get used repeatedly, the additional cost increases until eventually a counter-clockwise route is cheaper. A switch from clockwise to counter-clockwise then results in a jump in true cost that is proportional to the size of the ring.

The measured difference of each polymorphism from the previous (Δ), the diameter of each polymorphism, and the average number of hops for the commodities of

each polymorphism were recorded for all 30 random test cases of each configuration. The averages of these metrics are tabulated in Tables 22-25. The average Δ decreases as the number of nodes increases, for all configurations. For most of the configurations, the average Δ decreases as the number of commodities per node increases; however, it does not appear to be influenced by the number of interfaces per node. The average diameter tends to increase with respect to the number of nodes and decrease with respect to the number of interfaces per node. There may be a slight increase with respect to the number of commodities per node. Finally, the average number of hops increases with respect to the number of nodes and decreases with respect to the number of interfaces per node. The average number of hops does not appear to be influenced by the number of commodities per node.

Table 22: Other PNP results for 5-node configurations

Config.	Average Δ	Average Diameter	Average # of Hops
5N1C1I	0.32659037	2.78	1.814
5N1C2I	0.392781852	2.266666667	1.628666667
5N1C3I	0.395125185	2.176666667	1.593333333
5N1C4I	0.379387407	1.89	1.508
5N2C1I	0.314182222	3.023333333	1.811
5N2C2I	0.38457963	2.576666667	1.668666667
5N2C3I	0.39370037	2.39	1.626333333
5N2C4I	0.389385185	2.263333333	1.593
5N3C1I	0.285052963	2.996666667	1.788423333
5N3C2I	0.34692	2.54	1.653093333
5N3C3I	0.349472593	2.286666667	1.583996667
5N3C4I	0.341362593	2.2	1.565306667

Table 23: Other PNP results for 10-node configurations

Config.	Average Δ	Average Diameter	Average # of Hops
10N1C1I	0.175595556	4.793333333	2.686666667
10N1C2I	0.227894815	3.636666667	2.252333333
10N1C3I	0.21799963	3	2.014666667
10N1C4I	0.209883704	2.843333333	1.916666667
10N2C1I	0.166133704	5.25	2.694
10N2C2I	0.205910741	3.853333333	2.2535
10N2C3I	0.199552963	3.373333333	2.074666667
10N2C4I	0.192773333	3.123333333	1.9675
10N3C1I	0.160722593	5.5	2.783933333
10N3C2I	0.182757778	4.016666667	2.28145
10N3C3I	0.171428889	3.406666667	2.054923333
10N3C4I	0.167958148	3.07	1.9542

Table 24: Other PNP results for 15-node configurations

Config.	Average Δ	Average Diameter	Average # of Hops
15N1C1I	0.130969259	6.823333333	3.615143333
15N1C2I	0.164442593	4.35	2.63778
15N1C3I	0.155179259	3.676666667	2.300876667
15N1C4I	0.147706667	3.27	2.132676667
15N2C1I	0.117744444	7.293333333	3.64165
15N2C2I	0.143535556	4.716666667	2.657763333
15N2C3I	0.139707037	3.903333333	2.387923333
15N2C4I	0.132989259	3.57	2.213246667
15N3C1I	0.118127407	7.573333333	3.543036667
15N3C2I	0.124762963	4.71	2.644603333
15N3C3I	0.119145185	3.923333333	2.37531
15N3C4I	0.111257037	3.513333333	2.17035

Table 25: Other PNP results for 20-node configurations

Config.	Average Δ	Average Diameter	Average # of Hops
20N1C1I	0.087745185	8.903333333	4.459623333
20N1C2I	0.137043333	5.353333333	3.096166667
20N1C3I	0.126212963	4.193333333	2.608333333
20N1C4I	0.120387407	3.76	2.397833333
20N2C1I	0.085454074	9.293333333	4.324306667
20N2C2I	0.113743704	5.463333333	3.044416667
20N2C3I	0.106434074	4.336666667	2.591166667
20N2C4I	0.104573704	3.946666667	2.459083333
20N3C1I	0.085228148	9.043333333	4.056736667
20N3C2I	0.101305556	5.62	3.056013333
20N3C3I	0.092512963	4.446666667	2.607216667
20N3C4I	0.088161481	3.963333333	2.419093333

Lastly, an example of Δ by polymorphism for configuration 5N3C2I is given in Figure 38. The graph consists of 30 panels, each panel corresponding to one of the 30 test cases for the configuration. The vertical axis represents the Δ measurement and the horizontal axis represents the polymorphism number. Here, the ten polymorphisms are numbered from 0 to 9. Since polymorphism 0 does not have a previous polymorphism, no Δ value is plotted for it. Non-zero values for Δ indicate that a polymorphism is different than its predecessor, and larger values indicate a larger difference. None of the panels in Figure 38 have a zero value plotted, thus configuration 5N3C2I had no cases where any topology was repeated in consecutive iterations of the algorithm. Any kind of oscillation between polymorphisms is reflected as periodicity in the graphs. A lack of periodicity here indicates that most of the topologies generated are unique. A complete set of graphs of Δ by polymorphism for all configurations with 5-20 nodes is given in Appendix J.

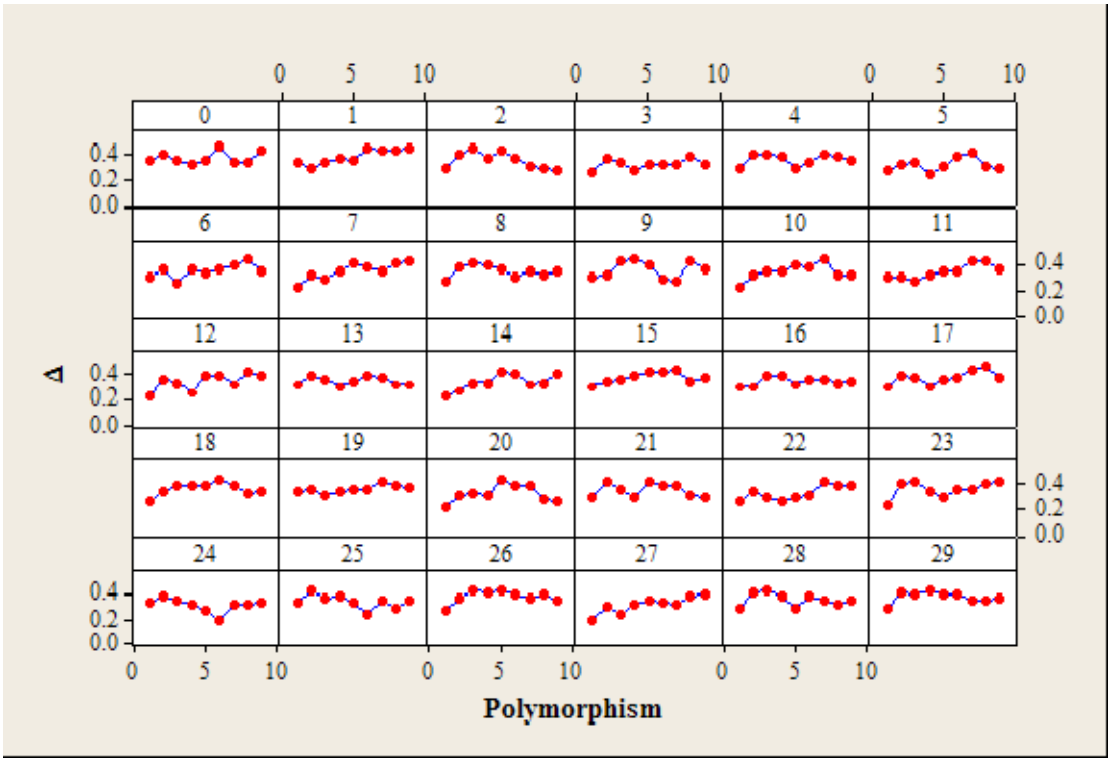

Figure 38: Plots of Δ by polymorphism for 5N3C2I

4.2.2 Networks of 25-40 Nodes

All configurations for networks with 25 to 40 nodes involving 1 or 2 interfaces types have been solved for 10 polymorphisms with 30 randomly generated input files for a full set of 300 polymorphisms. Due to extremely long running times, most of the other configurations were limited to only a few test cases. Six of the 40-node configurations and one of the 35-node configurations were not completed at all. Table 26 lists the number of polymorphisms that were completed for each of the configurations. The shorthand for each configuration is listed with the number of polymorphisms completed to its right.

Table 26: Number of polymorphisms completed for configurations of 25-40 nodes

Config.	# Poly.	Config.	# Poly.	Config.	# Poly.	Config.	# Poly.
25N1C1I	300	30N1C1I	300	35N1C1I	300	40N1C1I	300
25N1C2I	300	30N1C2I	300	35N1C2I	300	40N1C2I	300
25N1C3I	300	30N1C3I	30	35N1C3I	20	40N1C3I	0
25N1C4I	30	30N1C4I	10	35N1C4I	7	40N1C4I	0
25N2C1I	300	30N2C1I	300	35N2C1I	300	40N2C1I	300
25N2C2I	300	30N2C2I	300	35N2C2I	300	40N2C2I	300
25N2C3I	300	30N2C3I	30	35N2C3I	10	40N2C3I	0
25N2C4I	50	30N2C4I	10	35N2C4I	1	40N2C4I	0
25N3C1I	300	30N3C1I	300	35N3C1I	300	40N3C1I	300
25N3C2I	300	30N3C2I	300	35N3C2I	300	40N3C2I	300
25N3C3I	300	30N3C3I	30	35N3C3I	0	40N3C3I	0
25N3C4I	30	30N3C4I	10	35N3C4I	6	40N3C4I	0

Tables 27-30 show the 95% confidence intervals for the mean and median times in seconds required to generate each polymorphism for each configuration. The minimum and maximum solution times out of the generated topologies for each configuration are also given. The tables are shaded as in the previous subsection to ease comparison of configurations. In addition to the shading, configurations with fewer than 300 polymorphisms have the number completed in superscript.

It is hard to make comparisons in solution times when different numbers of polymorphisms are generated for different configurations. Also, the spread in solution times among the polymorphisms of a single configuration are very large. For example, in 35N3C4I, the fastest solution took just under two hours, while the slowest solution took more than 85 days. This makes the notion of an average solution time dubious. What is clear, both in this subsection and the previous, is the exponential growth in solution times with respect to the increased complexity of the base network.

Table 27: PNP time results for 25-node configurations

Config.	95% Confidence Interval for Mean Time in Seconds		95% Confidence Interval for Median Time in Seconds		Min. Time in Seconds	Max. Time in Seconds
25N1C1I	1.0184	1.0592	0.9910	1.0269	0.8140	2.2140
25N1C2I	32.707	43.311	20.636	26.883	2.903	468.499
25N1C3I	1323.4	1893.1	514.9	883.3	10.9	16777.2
$_{25N1C4I}^{30}$	32855	112853	5282	64570	59	394682
25N2C1I	2.6141	3.2717	1.9751	2.1824	1.5790	26.9410
25N2C2I	39.361	49.944	27.415	34.783	5.291	503.470
25N2C3I	1994.9	3174.3	539.7	882.6	24.2	51775.3
$_{25N2C4I}^{50}$	16035	54472	5098	20199	201	330146
25N3C1I	6.7123	8.2470	3.4928	4.6792	2.2980	44.4540
25N3C2I	48.630	58.069	39.882	48.314	6.134	288.084
25N3C3I	3549	7774	925	1472	28	272056
$_{25N3C4I}^{30}$	44680	165843	14786	73604	210	569782

Table 28: PNP time results for 30-node configurations

Config.	95% Confidence Interval for Mean Time in Seconds		95% Confidence Interval for Median Time in Seconds		Min. Time in Seconds	Max. Time in Seconds
30N1C1I	1.4333	1.4619	1.4016	1.4324	1.2660	2.2740
30N1C2I	124.38	257.88	52.71	81.86	8.06	9426.64
$_{30N1C3I}^{30}$	8530	43691	3311	21087	187	244854
$_{30N1C4I}^{10}$	25986	132166	5580	172557	54	177990
30N2C1I	4.4364	5.3939	3.1876	3.3114	2.6360	31.1190
30N2C2I	106.33	141.54	63.10	79.72	13.94	1302.26
$_{30N2C3I}^{30}$	1059	73567	2555	17165	101	473978
$_{30N2C4I}^{10}$	0	897539	18933	871666	3239	1654354
30N3C1I	8.0000	9.6210	5.9459	6.5789	4.0870	97.1820
30N3C2I	108.51	134.28	88.43	114.84	10.55	1024.74
$_{30N3C3I}^{30}$	15729	63577	1884	15636	63	192065
$_{30N3C4I}^{10}$	0	1925225	85607	1455458	41198	4092525

Table 29: PNP time results for 35-node configurations

Config.	95% Confidence Interval for Mean Time in Seconds		95% Confidence Interval for Median Time in Seconds		Min. Time in Seconds	Max. Time in Seconds
35N1C1I	2.9768	3.6391	2.7803	2.9023	2.2250	49.5100
35N1C2I	639.8	963.0	265.9	371.9	9.2	11622.8
$_{35N1C3I}^{20}$	0	1021051	28274	220710	7701	5571853
$_{35N1C4I}^{7}$	60456	3427479	381885	3592041	5057	5256748
35N2C1I	8.2356	9.3873	6.2912	6.9272	4.5420	31.5250
35N2C2I	524.17	771.77	231.79	329.76	11.85	8511.36
$_{35N2C3I}^{10}$	19781	165599	5267	190121	1489	284025
$_{35N2C4I}^{1}$	N/A	N/A	N/A	N/A	138942	138942
35N3C1I	12.352	14.925	9.521	11.169	6.794	133.301
35N3C2I	393.49	507.86	259.93	339.97	15.28	3124.15
$_{35N3C3I}^{0}$	N/A	N/A	N/A	N/A	N/A	N/A
$_{35N3C4I}^{6}$	0	5618167	20337	6568722	7146	7396868

Table 30: PNP time results for 40-node configurations

Config.	95% Confidence Interval for Mean Time in Seconds		95% Confidence Interval for Median Time in Seconds		Min. Time in Seconds	Max. Time in Seconds
40N1C1I	4.2310	4.7167	3.2750	3.3230	3.0570	15.2990
40N1C2I	1868.9	3480.5	600.4	934.3	8.8	82046.6
40N2C1I	6.3259	7.4164	5.9872	6.0900	5.4910	84.7150
40N2C2I	1096.4	1647.0	508.8	778.6	25.4	22684.5
40N3C1I	9.1202	9.5126	9.1306	9.2373	8.3640	37.9090
40N3C2I	875.0	1217.0	641.8	966.6	23.8	20374.4

As in the previous subsection, true solution costs across polymorphisms are examined using the percentage change from optimum. The values are computed as before, and the results are shown in Tables 31-34. Note that for configurations written in subscript, the number in superscript indicates the total number of polymorphisms found. For example, for $_{25N1C4I}^{30}$, there are three groups of ten polymorphisms for a total of 30.

In that case, the mean is the average of three percentage changes. For those with a superscript 10, the standard deviation cannot be calculated because there is only one value. For those with a superscript 1 or 0, none of the statistics are calculated.

Table 31: True cost results for 25-node configurations

Config.	Mean % Change in Cost	Standard Deviation	Min. % Change in Cost	Max. % Change in Cost
25N1C1I	19.68	7.63	8.67	38.01
25N1C2I	30.35	4.77	21.82	40.00
25N1C3I	28.86	5.09	20.56	40.63
25N1C4I^{30}	35.52	7.45	24.00	37.78
25N2C1I	21.94	8.52	3.35	40.43
25N2C2I	34.07	4.40	26.06	45.00
25N2C3I	30.06	3.84	23.39	38.79
25N2C4I^{50}	27.91	3.20	22.84	31.25
25N3C1I	18.50	6.66	6.01	38.02
25N3C2I	39.58	4.53	32.68	51.91
25N3C3I	33.94	3.11	25.53	40.34
25N3C4I^{30}	28.91	3.59	25.23	32.41

Table 32: True cost results for 30-node configurations

Config.	Mean % Change in Cost	Standard Deviation	Min. % Change in Cost	Max. % Change in Cost
30N1C1I	16.84	8.19	2.55	32.69
30N1C2I	32.33	5.63	24.82	47.41
30N1C3I^{30}	28.53	3.71	24.65	32.03
30N1C4I^{10}	34.51	N/A	34.51	34.51
30N2C1I	18.50	6.59	2.41	34.38
30N2C2I	35.75	4.47	27.56	45.68
30N2C3I^{30}	31.76	2.07	29.86	33.97
30N2C4I^{10}	24.39	N/A	24.39	24.39
30N3C1I	17.34	10.93	4.68	62.99
30N3C2I	40.14	3.89	32.53	50.63
30N3C3I^{30}	34.56	2.96	31.16	36.64
30N3C4I^{10}	23.78	N/A	23.78	23.78

Table 33: True cost results for 35-node configurations

Config.	Mean % Change in Cost	Standard Deviation	Min. % Change in Cost	Max. % Change in Cost
35N1C1I	17.01	7.66	2.26	31.07
35N1C2I	30.03	4.43	19.37	42.04
35N1C3I[20]	20.58	1.84	19.28	21.88
35N1C4I[7]	16.31	N/A	16.31	16.31
35N2C1I	16.60	6.35	6.90	33.82
35N2C2I	36.07	3.80	31.34	51.26
35N2C3I[10]	31.02	N/A	31.02	31.02
35N2C4I[1]	N/A	N/A	N/A	N/A
35N3C1I	15.31	5.96	2.62	24.24
35N3C2I	39.58	3.27	33.25	45.13
35N3C3I[0]	N/A	N/A	N/A	N/A
35N3C4I[6]	22.77	N/A	22.77	22.77

Table 34: True cost results for 40-node configurations

Config.	Mean % Change in Cost	Standard Deviation	Min. % Change in Cost	Max. % Change in Cost
40N1C1I	4.10	2.44	0.00	8.68
40N1C2I	30.96	3.76	21.97	39.11
40N2C1I	10.91	4.24	3.65	18.07
40N2C2I	37.05	4.83	28.18	47.08
40N3C1I	8.82	4.08	3.02	19.53
40N3C2I	41.60	3.35	34.33	49.77

There does not appear to be any clear patterns in solution costs correlating to the number of nodes, the number of commodities per node, or the number of interfaces per node. Perhaps the most interesting result in these tables is the minimum percentage change in cost of 0% for configuration 40N1C1I. This means there was a test case where all ten polymorphism had exactly the same cost. This does not necessarily mean that all

ten topologies are identical. However, upon closer inspection, the potential-adjacency matrix in one of the input files for 40N1C1I forms a tree. Hence, there is only one possible topology that can be made.

The largest maximum percentage change in cost was 62.99% for 30N3C1I. So, the observation still holds that the MILP approach to the PNP produces polymorphisms within 75% of optimum, at least for the potential-adjacency matrices used here.

The average Δ, diameter, and number of hops for the various configurations are tabulated in Tables 35-38. The average Δ does not exist for 35N2C4I because only one polymorphism was generated. Similarly, there is no average Δ, diameter, or number of hops for 35N3C3I since no polymorphisms were generated for it. These averages are taken across all polymorphisms for a configuration, not in groups of ten as was done for the true cost results above.

Table 35: Other PNP results for 25-node configurations

Config.	Average Δ	Average Diameter	Average # of Hops
25N1C1I	0.087516296	9.836666667	4.780966667
25N1C2I	0.110172593	5.536666667	3.164133333
25N1C3I	0.106053333	4.536666667	2.7852
$_{25N1C4I}^{30}$	0.094644444	4	2.448
25N2C1I	0.089023333	10.62	4.783786667
25N2C2I	0.095329259	5.753333333	3.2554
25N2C3I	0.088042963	4.58	2.776533333
$_{25N2C4I}^{50}$	0.083735556	4.1	2.5188
25N3C1I	0.073235926	9.973333333	4.379543333
25N3C2I	0.082624074	6.013333333	3.253183333
25N3C3I	0.07504	4.656666667	2.75129
$_{25N3C4I}^{30}$	0.068203704	4	2.478233333

Table 36: Other PNP results for 30-node configurations

Config.	Average Δ	Average Diameter	Average # of Hops
30N1C1I	0.059575926	12.19	5.5898
30N1C2I	0.097767037	6.02	3.480116667
$_{30N1C3I}^{30}$	0.0934	4.8	3.010033333
$_{30N1C4I}^{10}$	0.073971429	11.2	8.0467
30N2C1I	0.07024037	11.5	5.129233333
30N2C2I	0.081275185	6.213333333	3.485296667
$_{30N2C3I}^{30}$	0.076092593	4.966666667	2.9449
$_{30N2C4I}^{10}$	0.073044444	4.8	2.7516
30N3C1I	0.050427037	10.57666667	4.594466667
30N3C2I	0.073167778	6.553333333	3.537716667
$_{30N3C3I}^{30}$	0.064418519	5.2	2.948566667
$_{30N3C4I}^{10}$	0.061466667	4.4	2.731

Table 37: Other PNP results for 35-node configurations

Config.	Average Δ	Average Diameter	Average # of Hops
35N1C1I	0.050944074	12.79666667	6.063303333
35N1C2I	0.08598037	6.38	3.68876
$_{35N1C3I}^{20}$	0.082922222	5.05	3.13565
$_{35N1C4I}^{7}$	0.059233333	4.142857143	2.4
35N2C1I	0.056544444	12.39666667	5.48479
35N2C2I	0.072238889	6.523333333	3.65416
$_{35N2C3I}^{10}$	0.0643	4.6	2.9287
$_{35N2C4I}^{1}$	N/A	4	2.186
35N3C1I	0.041992593	11.18666667	4.878383333
35N3C2I	0.06449	6.703333333	3.671873333
$_{35N3C3I}^{0}$	N/A	N/A	N/A
$_{35N3C4I}^{6}$	0.04696	4.333333333	2.516

Table 38: Other PNP results for 40-node configurations

Config.	Average Δ	Average Diameter	Average # of Hops
40N1C1I	0.025561852	13.64666667	6.629453333
40N1C2I	0.079309259	6.963333333	3.97625
40N2C1I	0.036214444	13.31333333	5.626873333
40N2C2I	0.065314815	7.03	3.893213333
40N3C1I	0.024864074	12.2	5.02902
40N3C2I	0.058340741	7.08	3.870263333

The average Δ decreases as the number of nodes increases, for all configurations. As the number of commodities per node increases, the average Δ tends to decrease (for configurations with more than one interface type). As the number of interface types increases, the average Δ first increases then decreases. The average diameter appears to increase as the number of nodes increases; there are some anomalies among the configurations without a full set of test cases. As the number of commodities per node increases, the average diameter does not follow any clear pattern. The average diameter decreases as the number of interface types increases, the only exception being between 30N1C3I and 30N1C4I. The average number of hops appears to increase as the number of nodes increases; again, there are anomalies among the configurations without a full set of test cases. As the number of commodities per node increases, the average number of hops changes very little, except for configurations with one interface for which it tends to decrease. Finally, the average number of hops decreases as the number of interface types increases, the only exception being between 30N1C3I and 30N1C4I.

4.3 Polymorphic Networking Security

In this section the metric of average percentage active time (APAT) is presented and analyzed for the various network configurations outlined in Chapter Three (III). In addition to the APAT, counts of test cases whose polymorphisms contain at least one edge that is active 100% of the time are also given. As in the previous section, this section is split into results for networks of 5-20 nodes and networks of 25-40 nodes. This split is primarily made because some of the 25-40 node configurations were not able to be run to completion.

4.3.1 Networks of 5-20 Nodes

When looking at the APAT for the edges in the polymorphisms, there are clear trends. When the number of nodes and the number of commodities per node are fixed, the APAT decreases as the number of interfaces per node increases. As seen in Figure 39, the decrease appears to be inversely proportional to the number of interfaces per node. This makes sense because the traffic level remains constant as the number of available edges increases. The APAT for edges approaches zero. When a network has, on average, n edges between each pair of nodes, it is reasonable to expect that under light loads each edge is used $1/n$ of the time. If the network becomes more heavily loaded, the APAT increases since more edges are needed to accommodate the extra flow. This trend is noted in the next paragraph.

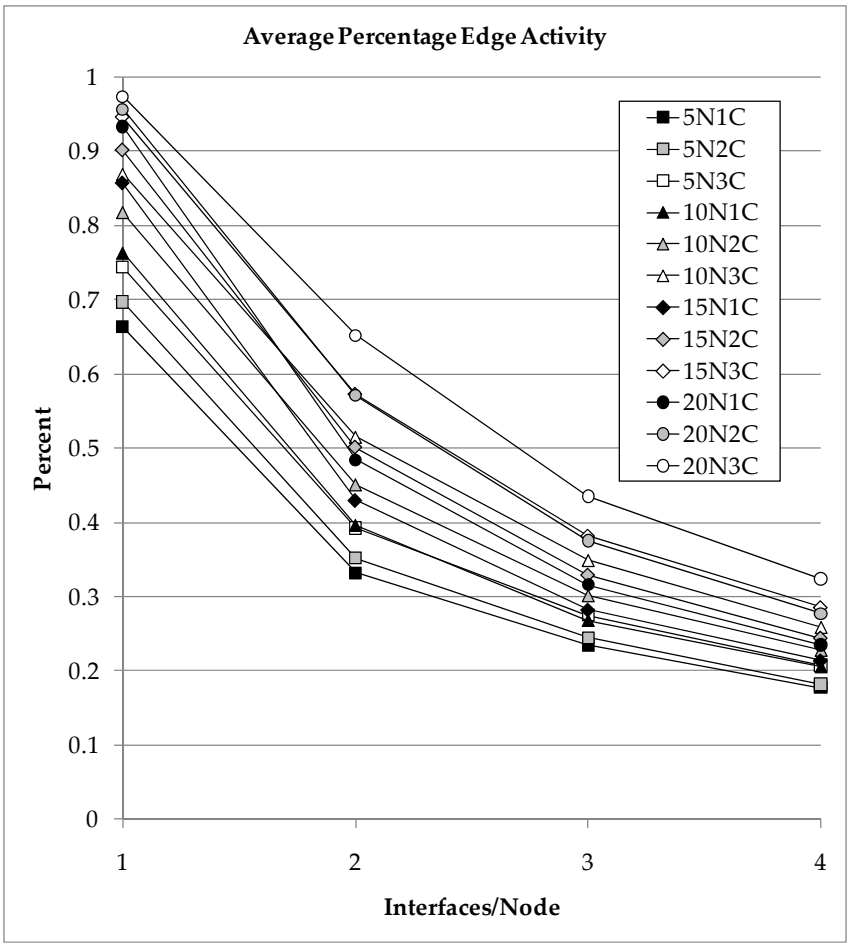

Figure 39: APAT vs. interfaces/node for PNP (5-20)N(1-3)C configurations

When the number of nodes and the number of interfaces per node are fixed, the APAT increases as the commodities per node increases. The increase, as seen in Figure 40, appears to be linear. Since the amount of traffic in the network is increasing while everything else is fixed, an increase in edge activity is expected. Since APAT cannot exceed 100%, the linearity cannot continue indefinitely.

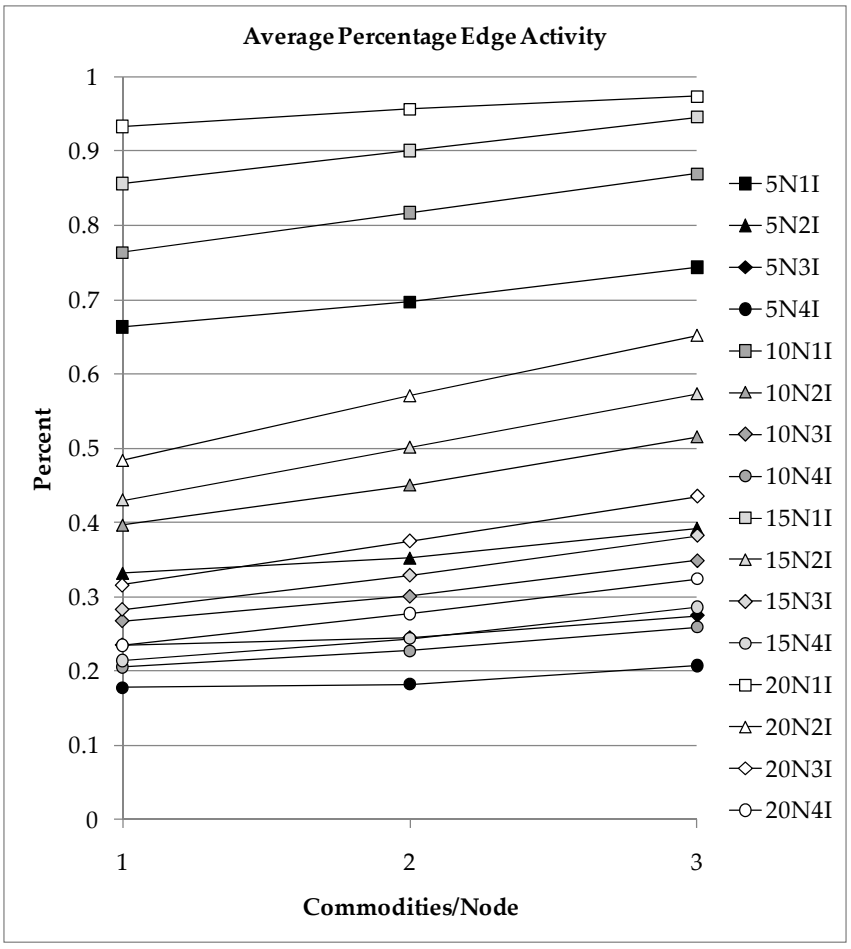

Figure 40: APAT vs. commodities/node for PNP (5-20)N(1-4)I configurations

Finally, when the number of commodities per node and the number of interfaces per node are fixed, the APAT increases as the number of nodes in the network increases. This is shown in Figure 41. In this case, the number of potential edges in the network increases with extra nodes, however the traffic also increases. The increase in traffic is evidently more influential than the increase in potential edges. It is unclear what type of function governs the growth.

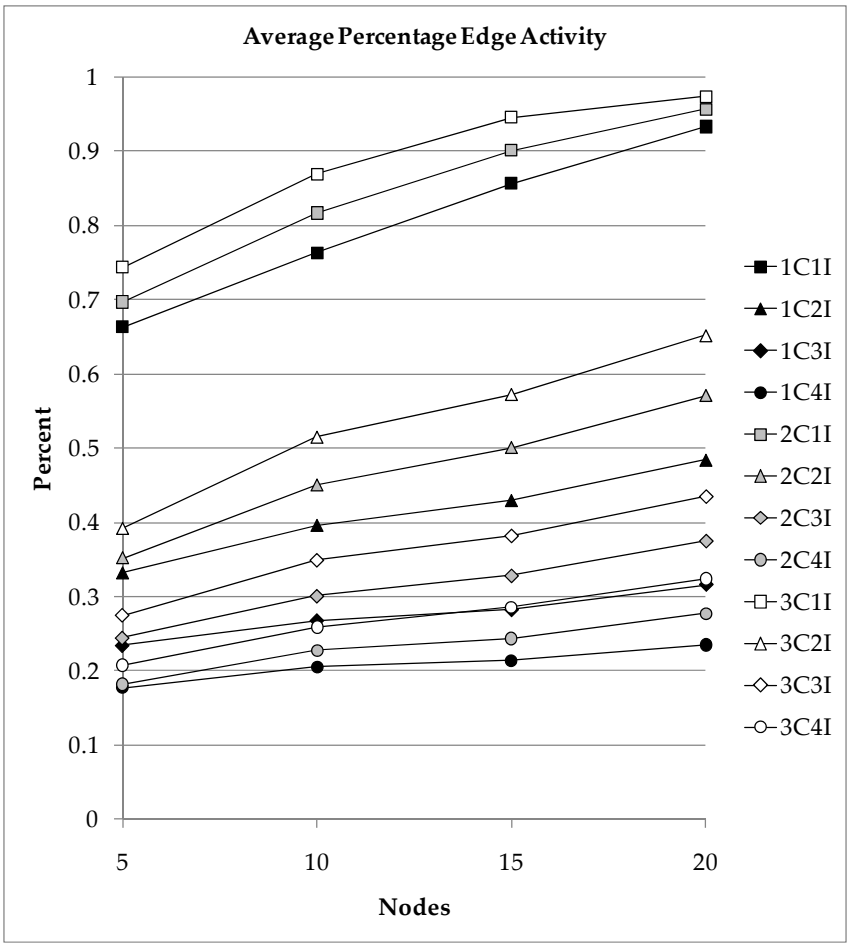

Figure 41: APAT vs. number of nodes for PNP (1-3)C(1-4)I configurations

The APAT results support the hypothesis that polymorphic networking strengthens a network against cyber attack. Moreover, a value can be given to the resistance. The value, of course, depends on the configuration of the network. For example, consider a network with 15 nodes, 2 interfaces per node, and 3 commodities per node. With polymorphic networking, an attacker eavesdropping on a link is expected to hear only about 58% of the data available to an attacker on a static network.

Over all 30 test cases for each configuration involving 4 interfaces per node, there was no single edge that was active for more than 80% of the polymorphisms. For

configurations with 5 nodes, only those with 1 interface per node had any edges that were active 100% of the time. For configurations with 10-20 nodes, all configurations with 1-3 interfaces per node, except 10N2C3I and 10N1C3I, had at least one case with edges that were active 100% of the time. For 1 interface per node, every test case had at least one 100% active edge. For 2 interfaces per node, the number of such test cases ranged from 3 to 26, and for 3 interfaces per node ranged from 1 to 5 test cases. Table 39 summarizes these results. As evidenced by these results, providing the ability to connect nodes in more than one way can significantly reduce the chances that attackers encounter a link that allows uninterrupted eavesdropping ability.

Table 39: Count of PNP cases (out of 30) containing at least one 100% active edge

	1I	**2I**	**3I**
20N3C	30	26	3
20N2C	30	26	3
20N1C	30	19	5
15N3C	30	18	2
15N2C	30	12	1
15N1C	30	13	4
10N3C	30	5	1
10N2C	30	3	
10N1C	30	6	
5N3C	18		
5N2C	17		
5N1C	11		

4.3.2 Networks of 25-40 Nodes

The trends in APAT for edges in polymorphisms of configurations with 25-40 nodes are very similar to those for 5-20 nodes. When the number of nodes and the number of commodities are fixed, the APAT decreases in inverse proportion to the number of interfaces per node. Recall from Table 26 on page 131 that only one

polymorphism was found for 35N2C4I. In a sense, the edges in that polymorphism are active 100% of the time, since there are no other polymorphisms to rotate among. For this reason, the point for 35N2C at 4 interfaces per node is not included in the graph in Figure 42. Any configurations for which no polymorphisms are found are also excluded.

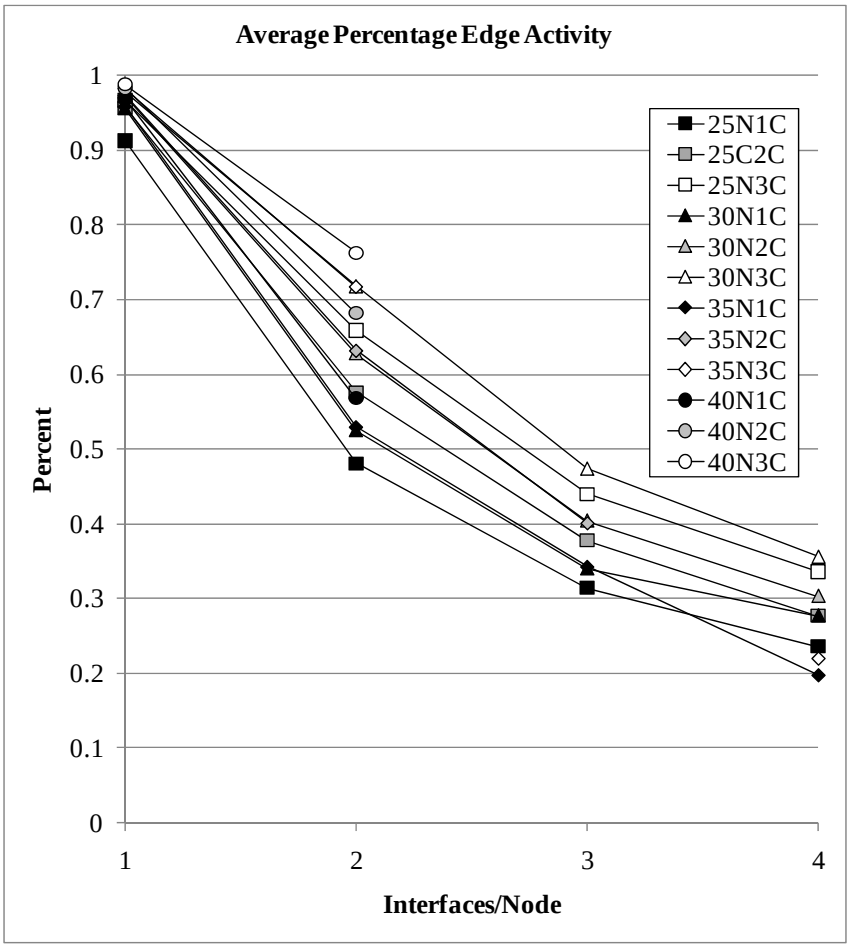

Figure 42: APAT vs. interfaces/node for PNP (25-40)N(1-3)C configurations

When the number of nodes and the number of interfaces per node are fixed, the APAT increases linearly as the commodities per node increases. In Figure 43, the point for 35N4I at 2 commodities per node is included even though only one polymorphism

was found. The value is put at 0.1, but based on results from other cases is probably closer to 0.209.

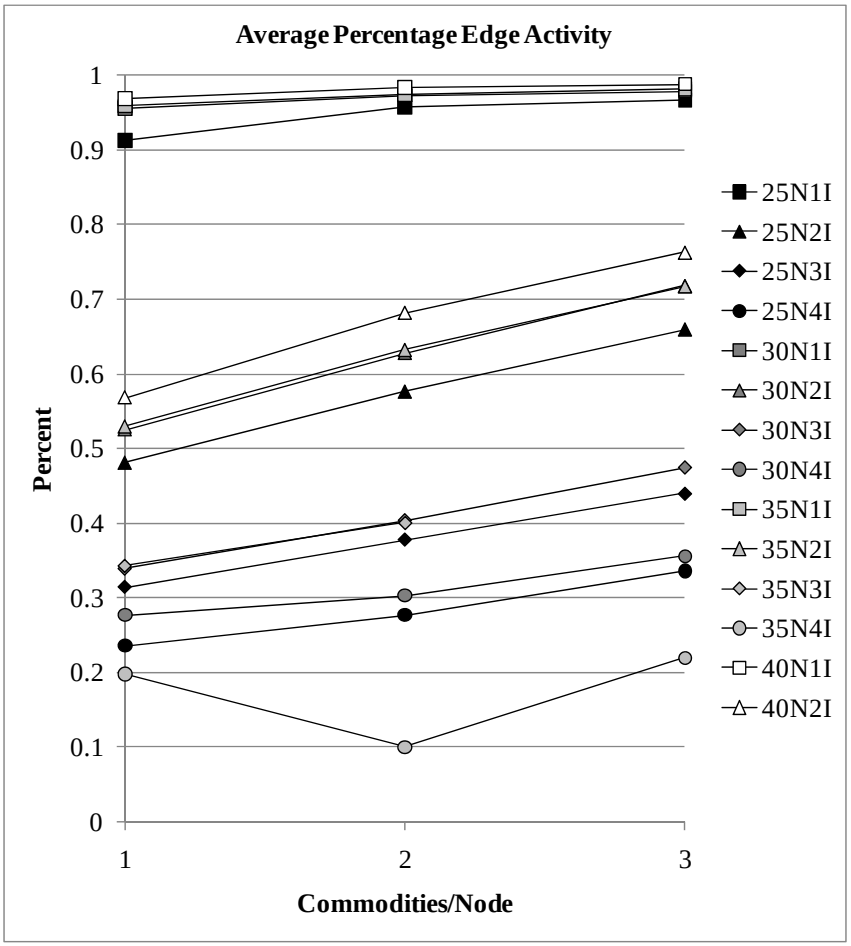

Figure 43: APAT vs. commodities/node for PNP (25-40)N(1-4)I configurations

In Figure 44, when the number of commodities per node and the number of interfaces per nodes are fixed, the APAT appears to increase as the number of nodes increases, with very little change from 30 to 35 nodes. There do appear to be some exceptions (1C4I, 2C4I, and 3C4I for 35 nodes) where APAT decreases. However, these points are less trustworthy because relatively few polymorphisms were generated for those configurations.

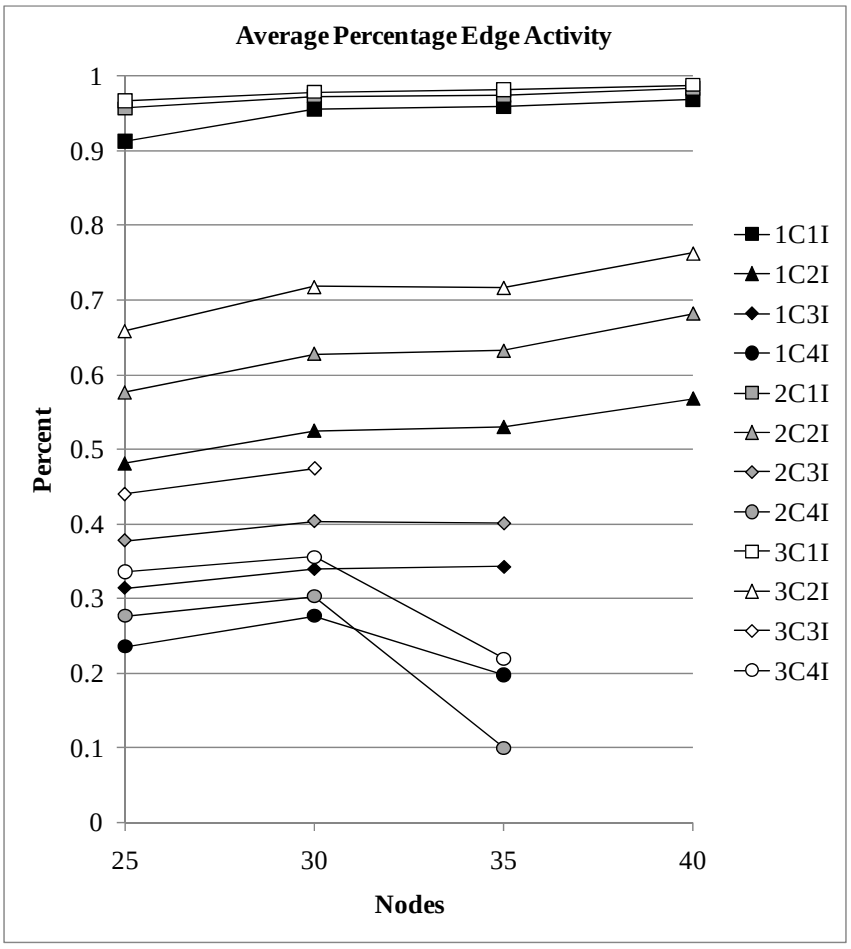

Figure 44: APAT vs. number of nodes for PNP (1-3)C(1-4)I configurations

Over all 12 configurations involving 1 interface per node, there were no test cases (out of 30) that did not have at least one edge that was active 100% of the time. For configurations with 2 interfaces per node, only 25N1C2I, 25N2C2I, 30N1C2I, 35N1C2I, and 40N1C2I had test cases where no edges were active 100% of the time. Respectively, the number of such test cases (out of 30) for each are 1, 1, 5, 7, and 10. The results for configurations with 3 and 4 interfaces per node are a little more complicated to report since a full set of 30 test cases were not run for most of the configurations. For 35N3C4I, only six polymorphisms for one test case were found. No edge was active in all six

polymorphisms. For 35N2C3I, one test case was solved with ten polymorphisms. There was at least one edge active in all ten. For 35N2C4I, only one polymorphism was found, so it does not make sense to count activity there. For the remainder of the configurations, very few test cases had any edges active 100% of the time. Table 40 summarizes the results. For entries with a '/', the first number indicates how many test cases had at least one 100% active edge, and the second number indicates how many test cases were run.

Table 40: Count of PNP cases containing at least one 100% active edge

	1I	**2I**	**3I**	**4I**
40N3C	30	30	N/A	N/A
40N2C	30	30	N/A	N/A
40N1C	30	29	N/A	N/A
35N3C	30	30	N/A	0/0.6
35N2C	30	30	1/1	N/A
35N1C	30	29	0/2	0/0.7
30N3C	30	30	2/3	0/1
30N2C	30	30	1/3	0/1
30N1C	30	25	0/3	0/1
25N3C	30	30	4/30	0/3
25N2C	30	23	4/30	0/5
25N1C	30	20	3/30	0/3

4.4 Summary

This chapter presented analysis and results for the simulations and experiments run during the course of this research. Three scenarios were simulated to support the hypothesis that having advanced knowledge of networking conditions gathered through the existence of an NTO process and using this information to pre-configure the network can improve the QoS of the GIG. The first scenario showed how the NTO process allows for increased *GIG-awareness*, which in turn leads to more informed decisions. In that scenario, an NTO directive eliminates a potential bottleneck in the network, resulting in

prevention of the loss of data from a high-priority source. The second scenario exhibits how topology control can play a role in the NTO process. In the second scenario, early recognition of aircraft trajectories enables network planners to direct a message intended for a single recipient over a specific path rather than broadcast to everyone in the region. Consequently, bandwidth is conserved, less unnecessary/redundant work is performed, and interference is avoided. The third scenario, designed by Göçmen [55], also examines the benefits of the NTO in the context of a CSAR mission. This scenario demonstrates how an improvement in latency may be possible by employing an NTO process.

In addition to the three scenarios, various aspects of the MILP formulation of the PNP were observed. The majority of the test configurations were run to completion using 30 randomly generated cases. As expected, solution times increase dramatically with respect to the number of nodes in the network. However, good results were generated for all configurations. Polymorphisms for most configurations displayed good variation with few cases of cyclic solutions. After the initial optimal solution, subsequent polymorphisms remained close to optimal in terms of diameter, average number of hops for commodities, and cost. The chapter ended with a look at the APAT results for the various configurations. The APAT results support the hypothesis that polymorphic networking strengthens a network against cyber attack.

The next chapter provides conclusions to be drawn from the results and analysis in this chapter. These conclusions are used to support the significance of researching the NTO process and urging its adoption. Finally, some topics for further research are recommended.

V. Conclusions and Recommendations

This chapter contains the conclusions generated from completing the four objectives of developing and describing the Network Tasking Order (NTO) process, producing scenarios in which the existence of an NTO process can be shown through simulation to improve the quality of service of a network, devising a polymorphic networking algorithm that takes its inputs from the NTO process and strengthens a network against cyber attack, and measuring the increased resistance of polymorphic networks to cyber attack. The conclusions are then used to explain the significance of the research. Finally, a list of recommendations for future research is given.

5.1 Conclusions of Research

The NTO process was proposed as a means of improving the quality of service and security of military networks by taking advantage of available foreknowledge of Global Information Grid (GIG) networking conditions. The first objective of this research was to develop and describe the NTO process. In doing so, sources of required information were identified, existing tools that can be utilized were explained, and the NTO's place in the Joint Air and Space Operations Center (JAOC) was defended. This proposed NTO process allows analysts to optimize the GIG and improve its security.

The second objective of this research was to exhibit scenarios in which the existence of an NTO process can be shown through simulation to improve the quality of service (QoS) of a network. Three different scenarios were provided to meet that objective. The results of each of the three scenarios showed a clear improvement to QoS when a NTO was utilized to enhance *GIG-awareness*.

The third objective was to develop a polymorphic networking algorithm. Erwin's mixed-integer linear programming (MILP) formulation for the Multi-commodity Capacitated Network Design Problem (MCNDP) was methodically corrected and modified using penalties for path reuse to solve the Polymorphic Networking Problem (PNP). The formulation was implemented successfully using the General Algebraic Modeling System (GAMS).

The last objective was to demonstrate that the polymorphic networks generated increase the resistance of a network to cyber attack. Although physical networks embodying the polymorphisms generated by the penalty approach have not been constructed, investigations into the average percentage active time (APAT) for edges in the networks show that employing polymorphism in networks with redundant connections can decrease an adversary's ability to eavesdrop significantly.

5.2 Significance of Research

The NTO process has been developed and described in far more detail than ever done previously. It has been shown to be a viable and beneficial tool for multiple reasons. First, the existence of an NTO process encourages better use of assets, which leads to improved QoS. With an NTO process, the GIG can be optimized in ways not previously possible. Shortcomings can be mitigated, bandwidth can be preserved, latency can be reduced, and interference can be avoided. Resources, such as frequency allocations or high-value aircraft, which are in short supply, may be applied more strategically. Further, having an NTO process can lead to better security for the GIG. Routed communications are less prone to interception than broadcast messages. By knowing in advance what

nodes and connections will be available in a network, polymorphic networking can be applied to make that network more resistant to cyber attack. In short, the warfighter benefits. The NTO process may also be applied in other special-purpose networks other than the GIG, such as the power grid and critical infrastructures.

As the Department of Defense moves more toward Net-Centric Warfare and Net-Centric Operations, having an NTO process in place now is crucial to permit the needed mechanisms to be established and to allow time for NTO content to evolve in a steady and controlled manner. In this way, a fully fledged NTO process can be ready to go before it is critically needed. This research has identified the mechanisms, tools, and sources required to incorporate an NTO production team into the roster of JAOC teams.

This research has also provided the proof-of-concept for a polymorphic networking algorithm with the goal of improving the security of the networks it is applied to. The results given here for networks of 5 to 40 nodes are optimal and provide a baseline for any future work to compare to. This leads naturally to the next section which provides ideas for future research on related topics.

5.3 Recommendations for Future Research

Because polymorphic networking is such a new topic, there are many ideas for future research. Since solutions found using the MILP formulation take an inordinate amount of time for larger networks, the logical next step is to look at heuristics for the PNP. The various heuristics that have been spawned to solve the MCNDP as described by Erwin can be easily adapted using the penalty approach for the PNP. Care is needed to ensure that the errors found in Erwin's formulation have not been propagated to these

heuristics as well. This idea alone can produce material for several Master's level theses. At the time of writing, Gabriel Greve at the Air Force Institute of Technology has already begun exploring the adaptation of heuristics for the disjoint paths problem to polymorphic networking.

Of course, other heuristic approaches to the PNP may prove fruitful. A simple stochastic approach is to start with a full collection of edges, removing edges at random. As each edge is removed, the resulting network is tested for feasibility of network flow. If the result is infeasible, the edge is replaced. If the result is feasible, the next random edge is removed. The process stops once every edge has either been removed or tested for infeasibility. This approach has the benefit that multiple solutions can be generated simultaneously. The major drawback of the PNP algorithm in this dissertation, aside from long running times, is that it must run linearly. One solution must be found before the next one can begin to be solved.

If heuristics are developed that can provide real-time solutions for moderately sized networks, it would be interesting to pair polymorphic networking with a network intrusion device. If the network intrusion device detects invasive activity on a particular link, the cost of that link can be increased to such a degree that future topologies will not have any traffic routed over that edge. The edge can also be removed from the potential adjacency matrix.

Another recommendation is to look into combining the polymorphic networking approach of this research with the dynamic network address translation approach of Kewley, et al. in [41]. The two approaches should complement each other to provide

additional security to the network. In order to achieve this pairing, it is necessary to build a physical network that can embody both approaches and be subjected to a red force attack.

Another interesting topic is to examine the effects of the polymorphic networking algorithm when faced with a variety of network types. The input files used here basically describe a random graph. If the adjacency percentage is set to 100%, the product is a complete graph. It would be interesting to see what kinds of results are obtained when presented with an exponential or scale-free network, for instance. Under what conditions does polymorphic networking give the best results and the worst results?

In regards to the NTO process, another venue where sufficient advance information is available for network planners to work with is the airline industry. The schedules and routes for regular flights are known months ahead of time. Any kind of mobile communications network that uses passenger planes for a backbone can surely benefit from the optimization and security that an NTO-like process provides. Research into how cancellations, delays, and rerouted flights should be handled to avoid negative impacts to the network needs to be performed.

5.4 Summary

Chapter Five (V) has discussed the conclusions and significance of this research. The NTO process can improve the QoS and security of military networks by enhancing *GIG-awareness* which facilitates the application of topology control and polymorphic networking. Several avenues of potential future research were also recommended.

Appendix A. TCNO Example

Time Compliance Network Orders (TCNO) are downward-directed operations, security, or configuration management-related orders that provide a standardized mechanism to issue one "order" to responsible Air Force agencies, directing how to operate and make changes to the Air Force Enterprise Network [65:17]. The following sample illustrates the format for a manually generated TCNO [65:59-60].

CLASSIFICATION: UNCLASSIFIED

RELEASE TIME: 03/09/2004 7:20:14:AM CST

TCNO TRACKING NUMBER: TCNO AFNOSC 2004-069-001

ORIGINATING AGENCY: AFNOSC

PRIORITY: Critical

SUBJECT: ASN.1 Vulnerability Could Allow Code Execution, MS 04-007

MISSION IMPACT: System Compromise

EXECUTIVE SUMMARY:

1. Summary

1.1. A vulnerability in the Microsoft ASN.1 Library could allow...

2. Implementation Details

2.1. Affected platforms, operating systems, applications, and versions:

Microsoft Windows NT Server 4.0 Service Pack 6a
Microsoft Windows NT Server 4.0 Terminal Server Edition Service Pack 6

2.2. Countermeasure implementation instructions

2.2.1. Users should download patches from ENOSC's website and follow the posted installation instructions.

2.3. Estimated downtime required to implement the countermeasures:

2.3.1. Manually per system: 5 to 20 minutes

2.3.2. Automated update system (e.g., SMS): 30 minutes per thousand clients.

2.4. Risks associated with non-compliance: Remote Code Execution

ACTION: Apply fix action specified in paragraph 2.2 to classified and unclassified systems to comply with this TCNO.

REMARKS:

1. Statistical Reporting

1.1. Report statistics in accordance with AFI 33-138, Chapter 3.

1.2. Discontinue status updates once compliance is reached and reported to the AFNOSC.

REPORTING REQUIREMENTS:

1. All organizations with eTANG capability will report compliance via eTANG. Compliance will be recorded in the appropriate boxes within the STATISTICS TAB.

2. For organizations lacking eTANG, reporting to the AFNOSC must be accomplished via SIPRNET E-mail message with the subject line of “COMPLIANCE STATISTICS FOR TCNO AFNOSC 2004-069-001”

RECEIPT ACKNOWLEDGMENT REQUIRED DATE: 17 Feb 04

COMPLIANCE REQUIRED DATE: 23 Feb 04

STATISTICS REQUIRED DATE: 23 Feb 04

POC INFORMATION: AFNOSC Crew Commander, mailto:afnosc@barksdale.af.mil, DSN 781-1043

REFERENCES: MS 04-007

Appendix B. C4 NOTAM Example

"Command, Control, Communications, and Computers Notice to Airmen (C4 NOTAM) are closely related to TCNOs with the primary difference being that they are informative in nature and are not used to direct actions" [65:33]. There are four types of C4 NOTAMs: Informative, Scheduled Event, Unscheduled Event, and Summary. The following sample illustrates the format for a manually generated Informative C4 NOTAM [65:69].

CLASSIFICATION: UNCLASSIFIED

RELEASE TIME: 03/09/2004 7:20:14:AM CST

TRACKING NUMBER: C4-N AFNOSC 2004-052-001

ORIGINATING AGENCY: AFNOSC

TYPE: Informative

PRIORITY: Low

SUBJECT: Port 3531 AFIN Block

MISSION IMPACT: Loss of Port 3531 communications prevents remote administration of non-critical legacy systems x and y. Nearby administrators still able to physically access system and accomplish mission.

EXECUTIVE SUMMARY: In response to NSIRC-047-04 addressing Peer to Peer (P2P) activity from military hosts, AFNOSC will implement a block for port 3531...

ACTION: Exemption requests should be directed to the AFNOSC...

REMARKS:

REPORTING REQUIREMENTS: None

RECEIPT RESPONSE REQUIRED DATE: None

COMPLIANCE REQUIRED DATE: None

STATISTICS REQUIRED DATE: None

POC INFORMATION: AFNOSC Crew Commander, afnosc@barksdale.af.mil, DSN 781-1043

REFERENCES: NSIRC-047-04

Appendix C. ATO Message Example

The following is an example ATO message taken from the United States Message Text Format (USMTF) Message Browser Help, 2004 Baseline edition [28]. The 69 characters per line maximum is violated here to conserve space and improve readability.

```
EXER/DESERT WIND//
MSGID/ATO/USCENTCOM/ATO A/OCT/CHG/1//
AKNLDG/YES//
TIMEFRAM/FROM:010600ZOCT1998/TO:020559ZOCT1998/ASOF:302100ZSEP1998//
TSKCNTRY/US//
SVCTASK/N//
TASKUNIT/CVN68 VA-165/ICAO:NMTZ/00201-00320//
AMSNDAT/N/0111I/-/AN/MC/INT//
MSNACFT/3/ACTYP:A6E/TALON 11/2GBU/-/TN11/-/00121/B:20111/00122/B:30111/00123/-//
ARINFO/APPLE 20/4010A/B:34010/NAME:BLUE TRACK/200/ARCT:010815Z/NDAR:010845ZOCT
/KLBS:30.0/PFREQ:343.3/SFREQ:277.8/AE20/ACTYP:KC10/CDT/2/TNKR:1/18-81/2-2-3//
1MSNRTE
/NAME               /ENTRY TIME/ENTRY PT  /EXIT TIME/EXIT PT   /TAS
/BLUE 23            /010900ZOCT/ALFA      /011000Z  /CHARLIE   / 370//
9PKGDAT
/PKGID/UNIT               /MSNNO   /PMSN  /NO/ACTYPE   /ACSIGN
/AN   /CVN68 VA-165       /0111I   /INT   / 3/AC:A6E   /TALON 11
/AN   /CVN68 VAQ-138      /0171S   /EW    / 1/AC:EA6B  /CLAW 71
/AN   /CVN68 VF-24        /0131D   /ESC   / 2/AC:F14A  /BEAK 31//
GTGTLOC/P/TOT:011000ZOCT/NET:010955ZOCT/NLT:011005Z
/MAIN COMMAND CENTER/ID:N1234F12345AA001/CP/NORTH COMPLEX
/DMPIS:354738N0473815E/WE/257FT/A1497/1/DESTROY/AA137//
REQNO/1F785I//
CONTROLA/AWAC/DARKSTAR/PDESIG:GOLD/SDESIG:BLUE/DR01/NAME:GINGER//
TASKUNIT/CVN68 VAQ-138/ICAO:NMTZ/00301-00450//
POC/ZAPOLSKI/CDR/N61/LOC:NIMITZ/FRQ:243.0GHZ//
AMSNDAT/N/0171S/-/AN/-/EW/SEAD//
MSNACFT/1/ACTYP:EA6B/CLAW 71/HARM/PODS/-/-/-/B:20171/-/B:30171//
ARINFO/APPLE 20/4010A/B:34010/NAME:BLUE TRACK/200/ARCT:010815Z/NDAR:010845ZOCT
/KLBS:20.0/PFREQ:343.3/SFREQ:277.8/AE20/ACTYP:KC10/CDT/2/TNKR:2/18-81/2-2-3//
PKGCMD/AN/CVN68 VA-165/0111I/TALON 11/TN01//
AMSNLOC/010950ZOCT/011010ZOCT/SEIRAQ/270/1//
CONTROLA/AWAC/DARKSTAR/PDESIG:GOLD/SDESIG:BLUE/DR01/NAME:GINGER//
6EWDATA
/EMITTYP        /ELNOT  /FC/LOWFRQ       /UPFRQ        /EA-TECQ
/TYPE:GCIRDR    /-      /ET/ F:365.798GHZ/ F:650.477GHZ/INCDRGPO//
TASKUNIT/CVN68 VF-24/ICAO:NMTZ/00401-00510//
AMSNDAT/N/0131D/-/AN/-/ESC//
MSNACFT/2/ACTYP:F14A/BEAK 31/2P2S2/FAMMO/BK31/25/00125/B:20131/00126/B:30131//
ARINFO/GRAPE 11/4011A/B:34011/NAME:ORANGE TRACK/200/ARCT:010815Z/NDAR:010845ZOCT
/KLBS:20.0/PFREQ:323.3/SFREQ:242.8/GE11/ACTYP:KC135/CDT/3/TNKR:1/29-92/3-3-4//
MSNACFT/2/ACTYP:F14A/BEAK 33/4B2S2/BEST/BK33/26/00126/B:20133/00127
/B:30133//
ARINFO/GRAPE 11/4011A/B:34011/NAME:ORANGE TRACK/200/ARCT:010815Z/NDAR:010845ZOCT
/KLBS:20.0/PFREQ:323.3/SFREQ:242.8/GE11/ACTYP:KC135/CDT/3/TNKR:2/29-92/3-3-4//
URMKREF/A//
PKGCMD/AN/CVN68 VA-165/0111I/TALON 11//
AMSNLOC/010845ZOCT/011200ZOCT/SEIRAQ/310/1//
CONTROLA/AWAC/DARKSTAR/PDESIG:GOLD/SDESIG:BLUE/DR01/NAME:GINGER//
GENTEXT/UNIT REMARKS/A: FORCE PROTECTION FOR VA-165 MSN NO. 0111I//
TASKUNIT/CVN68 VS-33/ICAO:NMTZ//
AMSNDAT/N/4020N/-/-/-/SUCAP/AR//
MSNACFT/1/ACTYP:S3B/PLUG 20/4MK20/AREF/-/-/-/B:24020/-/B:34020//
AMSNLOC/010900ZOCT/010930ZOCT/JUDY/210//
CONTROLA/AWAC/DARKSTAR/PDESIG:GOLD/SDESIG:BLUE/DR01/NAME:GINGER//
AMSNLOC/010930ZOCT/011000ZOCT/CVOA4/180//
CONTROLA/AWAC/DARKSTAR/PDESIG:GOLD/SDESIG:BLUE/DR01/NAME:GINGER//
```

```
TASKUNIT/LAKE CHAMPLAIN//
MTGTLOC/P/22336651/TOT:010930ZOCT/NET:010915Z/NLT:010945Z/ID:TAA0123456
/BLDG/T/3//
MTGTLOC/A/11223344/TOT:010945ZOCT/NET:010930Z/NLT:011000Z/ID:TAA7890123
/BLDG/T/3//
MTGTLOC/P/22336652/TOT:010930ZOCT/NET:010915Z/NLT:010945Z/ID:TAA0134567
/BLDG/T/3//
MTGTLOC/A/11223345/TOT:010945ZOCT/NET:010930Z/NLT:011000Z/ID:TAA4561230
/BLDG/T/3//
SVCTASK/M//
TASKUNIT/3MAW/ICAO:RDAH/04521-34552//
AMSNDAT/N/0211D/-/-/-/FAC//
MSNACFT/1/ACTYP:FA18C/SABER 11/2S2WG/TFLIR/-/-/-/B:20211/-/B:30211//
ARINFO/GRAPE 11/4011A/B:34011/NAME:ORANGE TRACK/210/ARCT:010915Z/NDAR:010945ZOCT
/KLBS:16.0/PFREQ:323.3/SFREQ:242.8/GE11/ACTYP:KC135/CDT/1/TNKR:1/29-92/3-3-4//
AMSNLOC/010900ZOCT/012020ZOCT/CHEVY/170/2/ELL:100M-150M-240.0//
REQNO/2M438C//
CONTROLA/AWAC/DARKSTAR/PDESIG:GOLD/SDESIG:BLUE/DR01/NAME:GINGER//
CONTROLA/DASC/VAGABOND/PFREQ:121.5//
ASUPTBY/SEAD/3MAW/0151H/HAMMER 51/HR51//
8FACSCHD
/MSNNO   /ATKACCS     /ATIME   /NO/ACTYPE   /WPNTY
/0271A   /LUSTY 71    /010915Z / 2/AC:AV8B  /4MK82//
AMSNDAT/N/0151H/-/-/-/SEAD//
MSNACFT/2/ACTYP:FA18/HAMMER 51/2HARM/2S2WG/HR51/-/00112/B:20151/00113/B:30151//
ARINFO/GRAPE 11/4011A/B:34011/NAME:ORANGE TRACK/200/ARCT:010815Z/NDAR:010845ZOCT
/KLBS:30.0/PFREQ:323.3/SFREQ:242.8/GE11/ACTYP:KC135/CDT/3/TNKR:3/29-92/3-3-4//
AMSNLOC/010950ZOCT/012010ZOCT/CHEVY/270//
CONTROLA/AWAC/DARKSTAR/PDESIG:GOLD/SDESIG:BLUE/DR01/NAME:GINGER//
ASUPTFOR/XCAS/2MAW VMA-542/0271A/LUSTY 71/LY71//
ASUPTFOR/FAC/3MAW/0211D/SABER 11//
TASKUNIT/2MAW VMA-542/ICAO:KNKT//
AMSNDAT/N/0271A/-/-/-/XCAS//
MSNACFT/2/ACTYP:AV8B/LUSTY 71/4MK82/BEST/-/-/-/B:30271/-/B:30272//
ARINFO/GRAPE 11/4011A/B:30115/NAME:ORANGE TRACK/200/ARCT:010900Z/NDAR:010930ZOCT
/KLBS:10.0/PFREQ:323.3/SFREQ:242.8/GE11/ACTYP:KC135/CDT/3/TNKR:3/29-92/3-3-4//
AMSNLOC/010915ZOCT/010945ZOCT/CHEVY/230/1//
REQNO/2M438C//
CONTROLA/AWAC/DARKSTAR/PDESIG:GOLD/SDESIG:BLUE/DR01/NAME:GINGER//
ASUPTBY/SEAD/3MAW/0151H/HAMMER 51/HR51//
FACINFOR/SABER 11/PFREQ:227.5/SFREQ:311.3/-/NAME:CHEVY/1BN8MA//
TASKUNIT/2MAW 2DRPVCO/ICAO:KNJM/03215-54236//
AMSNDAT/F/0291B/-/-/-/REC//
MSNACFT/1/OTHAC:PIONER/SEEKER 91/FLIR/-/SR91/-/00114/B:30271//
RECCEDAT/P/PRI:2/010700ZOCT/NET:010645ZOCT/NLT:010715Z/LTIOV:011345ZOCT1998
/FLR/PINPT/FL/IMQ:V/CM:C/C/ANA/-/12/12-1/12-2/12-3/12-4//
PTRCPLOT/LATS:300105N0803428W/NAME:BLUE RIVER BRIDGE/RAD:1NM/150FT/WE//
IMDATLNK/DESIG:PINK/CATCHER 23/LATM:3010N07930W/-/010700Z/010900Z//
REQNO/4M201//
TASKUNIT/2MAW HMM-365/ICAO:KNRR//
AMSNDAT/N/020001/-/-/-/HLOG//
MSNACFT/4/ACTYP:CH46E/PEDRO 01/-/-/-/-/-/B:30001/-/B:30004//
AMSNLOC/010900ZOCT/011000ZOCT/-/15/1/-/NAME:NEW RIVER/NAME:CAMP DAVIS//
REQNO/P297//
CONTROLA/DASC/VAGABOND/PFREQ:121.5//
ASUPTBY/ESC/2MAW HMLA-167/020005/SNAKE 05/SE05//
LANDSTS/NAME:NEW RIVER/-/-/-/-/CONTACT:RIVER 11/RR11/PDESIG:DELTA//
LANDSTS/NAME:CAMP DAVIS/HOT/301630Z/PANELS/ORANGE/CONTACT:DAVIS 11
/-/PDESIG:DELTA//
TASKUNIT/2MAW HMLA-167/ICAO:KNRR//
AMSNDAT/N/020005/-/-/-/AH/ESC//
MSNACFT/2/ACTYP:AH1W/SNAKE 05/FAMMO/2.75 ZUNI/-/-/-/B:30005/-/B:30006//
ESCDATA/020001/04/ACTYP:CH46E/HLOG/-/010900Z/STANAME:NEW RIVER/011000Z
/STPNAME:CAMP DAVIS/PEDRO 01/-/PFREQ:156.5/SFREQ:121.5//
AMSNLOC/010900ZOCT/011000ZOCT/-/15/1/-/NAME:NEW RIVER/NAME:CAMP DAVIS//
REQNO/P297//
CONTROLA/DASC/VAGABOND/PFREQ:121.5//
LANDSTS/NAME:NEW RIVER/-/-/-/-/CONTACT:RIVER 11/RR11/PDESIG:DELTA//
LANDSTS/NAME:CAMP DAVIS/HOT/301630Z/PANELS/ORANGE/CONTACT:DAVIS 11
/-/PDESIG:DELTA//
```

```
SVCTASK/F//
TASKUNIT/4FW/ICAO:KMFG/03567-03677//
POC/SMITH/LTC/21FS/LOC:KMFG/TEL:602-555-3377//
AMSNDAT/N/0101E/-/AAF/MC/INT//
MSNACFT/4/ACTYP:F15E/LIGHTNING 01/4XRF3/BEST/LG01/25/00111/B:20101
/00112/B:31234//
ARINFO/APPLE 20/4010A/B:34010/NAME:BLUE TRACK/200/ARCT:010830Z/NDAR:010900ZOCT
/KLBS:30.0/PFREQ:343.3/SFREQ:277.8/AE20/ACTYP:KC10/BOM/2/TNKR:2/18-81/2-2-4//
9PKGDAT
/PKGID/UNIT               /MSNNO   /PMSN  /NO/ACTYPE   /ACSIGN
/AAF  /4FW                /0101E   /INT   / 4/AC:F15E  /LIGHTNING 01
/AAF  /1FW                /0121C   /FCAP  / 4/AC:F15C  /EAGLE 21//
GTGTLOC/P/TOT:011000ZOCT/NET:010955ZOCT/NLT:011005Z/MAIN COMMAND
CENTER/ID:N1234F12345AA001/CP/SOUTH COMPLEX
/DMPIS:354738N0473815E/WE/257FT/A1497/1/DESTROY/AA135//
REQNO/1F785I//
CONTROLA/AWAC/DARKSTAR/PDESIG:GOLD/SDESIG:BLUE/DR01/NAME:GINGER//
PGMINFO/LC:1234//
GTGTLOC/A/TOT:011020ZOCT/NET:010101ZOCT/NLT:011030Z/COMMAND BUNKER
/ID:N1234F12367AB001/CP/CONCRETE BUNKER/DMPIS:354740N0473827E/WE
/257FT/A1499/1/DESTROY/AA235//
REQNO/1F796I//
CONTROLA/AWAC/DARKSTAR/PDESIG:GOLD/SDESIG:BLUE/DR01/NAME:GINGER//
TASKUNIT/1FW/ICAO:LLKA//
AMSNDAT/N/0121C/-/AAF/-/BRCAP/-/-/DEPLOC:LBNA/010715ZOCT/ARRLOC:LLKA
/010815ZOCT//
MSNACFT/4/ACTYP:F15C/EAGLE 21/2IR6RK/BEST/-/27/-/B:20121/-/B:31241//
ARINFO/APPLE 20/4010A/B:34010/NAME:BLUE TRACK/200/ARCT:010845Z/NDAR:010900ZOCT
/KLBS:30.0/PFREQ:343.3/SFREQ:277.8/AE20/ACTYP:KC10/BOM/2/TNKR:2/18-81/2-2-4//
URMKREF/A//
PKGCMD/AAF/4FW/0101E/LIGHTNING 01/LG01//
AMSNLOC/011000ZOCT/011200ZOCT/SEIRAQ/260//
CONTROLA/AWAC/DARKSTAR/PDESIG:GOLD/SDESIG:BLUE/DR01/NAME:GINGER//
GENTEXT/UNIT REMARKS/A:BARCAP FOR PACKAGE AAF//
TASKUNIT/465ARS/ICAO:KKLS/00356-27453//
AMSNDAT/N/4011A/-/-/-/AR//
MSNACFT/1/ACTYP:KC135R/GRAPE 11/CLD/-/GE11/-/00113/B:24011/-/B:32221//
AMSNLOC/010800ZOCT/011200ZOCT/ORANGE TRACK/200//
REFTSK/CDT/KLBS:50.0/KLBS:20.0/PFREQ:323.3/SFREQ:242.8/29-92/3-3-4//
5REFUEL
/MSNNO   /RECCS       /NO/ACTYPE   /OFLD     /ARCT   /SEQ  /TYP  /ARS
/0131D   /BEAK 31     / 2/AC:F14A  / KLB:20.0/010815Z/  A:1/A:JP8/CDT
/0131D   /BEAK 33     / 2/AC:F14A  / KLB:20.0/010815Z/  A:1/A:JP8/CDT
/0151H   /HAMMER 51   / 2/AC:FA18  / KLB:30.0/010815Z/  A:3/A:JP8/CDT
/0211D   /SABER 11    / 1/AC:FA18C / KLB:16.0/010915Z/  A:1/A:JP8/CDT
/0271A   /LUSTY 71    / 2/AC:AV8B  / KLB:10.0/010900Z/  A:1/A:JP8/CDT//
CONTROLA/AWAC/DARKSTAR/PDESIG:GREEN/SDESIG:WHITE/DR01/NAME:JIM//
TASKUNIT/22ARW/ICAO:KNCY/00247-12345//
AMSNDAT/N/4010A/-/-/-/AR//
MSNACFT/1/ACTYP:KC10/APPLE 20/CWT/-/AE20/27/00114/B:24010/-/B:34440//
AMSNLOC/010800ZOCT/011200ZOCT/BLUE TRACK/200//
REFTSK/CDT/KLBS:105.0/-/PFREQ:343.3/SFREQ:277.8/18-81/2-2-4//
5REFUEL
/MSNNO   /RECCS       /NO/ACTYPE   /OFLD     /ARCT   /SEQ  /TYP  /ARS
/0111I   /TALON 11    / 3/AC:A6E   / KLB:30.0/010815Z/  A:1/A:JP8/CDT
/0171S   /CLAW 71     / 1/AC:EA6B  / KLB:20.0/010815Z/  A:2/A:JP8/CDT
/0101E   /LIGHTNING 01/ 4/AC:F15E  / KLB:30.0/010830Z/  A:2/A:JP8/BOM
/0121C   /EAGLE 21    / 4/AC:F15C  / KLB:30.0/010845Z/  A:2/A:JP8/BOM
/0222C   /HEAVY 01    / 1/AC:B52   / KLB:30.0/011000Z/  A:1/A:JP8/BOM//
CONTROLA/AWAC/DARKSTAR/PDESIG:GREEN/SDESIG:WHITE/DR01/NAME:JIM//
TASKUNIT/552ACW/ICAO:KNFA/02345-12342//
AMSNDAT/N/AFC002/-/-/-/AEW//
MSNACFT/1/ACTYP:E3A/SKYWATCH 43/-/-/SH43/35/00115/B:20123/-/B:30123//
AMSNLOC/010600ZOCT/011500ZOCT/CONTROL RACETRACK/350//
ASACSDAT/AWAC/DARKSTAR/AWACS/AEW/DR01/-/-/PDESIG:GREEN/SDESIG:WHITE//
7CONTROL
/MSNNO   /ACSIGN      /NO/ACTYPE   /MSNTY /TOSTA  /RIP
/0111I   /TALON 11    / 3/AC:A6E   /INT   /010930Z/2840N08040W
/0171S   /CLAW 71     / 1/AC:EA6B  /EW    /010945Z/2840N08040W
/0131D   /BEAK 31     / 4/AC:F14A  /ESC   /010940Z/2840N08040W
```

```
/4020N   /PLUG 20     / 1/AC:S3B   /SUCAP /010810Z/2840N08040W
/0211D   /SABER 11    / 1/AC:FA18C /FAC   /010920Z/2840N08040W
/0151H   /HAMMER 51   / 2/AC:FA18  /SEAD  /010925Z/2840N08040W
/4011A   /GRAPE 11    / 3/AC:KC135R/AR    /010800Z/2540N08240W
/4010A   /APPLE 20    / 2/AC:KC10  /AR    /010805Z/2540N08240W
/0101E   /LIGHTNING 01/ 4/AC:F15E  /INT   /011000Z/2650N08340W
/0121C   /EAGLE 21    / 4/AC:F15C  /FCAP  /010945Z/2750N08345W
/0222C   /HEAVY 01    / 1/AC:B52   /INT   /011100Z/3725N08921W
/0271A   /LUSTY 71    / 2/AC:AV8B  /XCAS  /010915Z/2845N08050W//
TASKUNIT/5BW/ICAO:KDZZ/02341-34561//
AMSNDAT/N/0221B/-/-/-/OTR/-/-/DEPLOC:KNFA/010800ZOCT/ARRLOC:KDZZ/011800ZOCT//
AMPN/MISSION IS TO DESTROY TWO ENEMY SHIPS ESCAPING FROM HARBOR//
MSNACFT/2/ACTYP:B52H/MAUL 01/BEST/-/ML01/-/00116/B:20124/00117/B:30124//
SHIPTGT/P/TOT:011030ZOCT/-/-/HOS/DD/-/-/1/-/SINK/292330Z
/LATM:0123N04525E/-/260/10//
SHIPTGT/A/TOT:011100ZOCT/-/-/HOS/FF/-/-/1/-/SINK/292330Z
/LATM:0123N04525E/-/359/10//
AMSNDAT/F/0222C/-/-/-/INT/-/-/DEPLOC:KNFA/011000ZOCT/ARRLOC:KDZZ/020630ZOCT//
MSNACFT/1/ACTYP:B52H/HEAVY 01/CM/-/-/-/-/B:20127/-/B:30127//
ARINFO/APPLE 20/4010A/B:34010/NAME:BLUE TRACK/200/ARCT:011000Z/NDAR:011015ZOCT
/KLBS:30.0/PFREQ:343.3/SFREQ:277.8/AE20/ACTYP:KC10/BOM/2/TNKR:2/18-81/2-2-4//
AMSNLOC/011045ZOCT/011115ZOCT/SWIRAQ/280//
MTGTLOC/P/0222D/TOT:011100ZOCT/NET:011050Z/NLT:011110Z
/ID:N1244F12467AB011/AFBASE/C/5/1/DESTROY//
MTGTLOC/P/0222E/TOT:011115ZOCT/NET:011050Z/NLT:011120Z
/ID:N1356F13567AB021/SSMHQ/C/3/2/DESTROY//
CONTROLA/AWAC/DARKSTAR/PDESIG:GREEN/SDESIG:WHITE/DR01/NAME:JIM//
TASKUNIT/314AW/ICAO:KDZZ//
AMSNDAT/N/7001/AMC:0994XZ/-/-/TAL//
MSNACFT/1/ACTYP:C130/PICKUP 01/PALLETS/-/-/-/-/B:20125/-/B:30125//
AIRMOVE/1/AMC0994XA285/KNFA/B/-/-/ONN/011130Z/R35667/1B2//
TRANSREQ/0994XA/ULN:178962A/01/-/-/2.0//
AIRMOVE/2/AMC0994XB285/KLKA/B/-/011230ZOCT/DNN/011345Z/R48721/1B2//
TRANSREQ/0994XB/ULN:178963B/-/02/-/1.5//
AIRMOVE/3/AMC0994XC285/KNJM/B/-/011510ZOCT/BNN/011720Z/R44221/1B2//
TRANSREQ/0994XC/ULN:178964C/-/03/-/.5//
AIRMOVE/4/AMC0994XD285/KNKT/B/-/011830ZOCT/DNN/011900Z/R48900/1B1//
TRANSREQ/0994XD/ULN:217863D/03/-/-/1.0/NO/K-LOADER//
AIRMOVE/5/AMC0994XE285/KNFA/B/-/012000ZOCT/TNN//
TASKUNIT/4DASC/NAME:BLUE LOCATION//
ASACSDAT/DASC/VAGABOND/G29 RADIO/CMD/-/-/-/PFREQ:121.5//
7CONTROL
/MSNNO   /ACSIGN      /NO/ACTYPE   /MSNTY /TOSTA  /RIP
/0211D   /SABER 11    / 1/AC:FA18C /FAC   /010900Z/2845N08045W
/020001  /PEDRO 01    / 4/AC:CH46E /HLOG  /010830Z/2830N08000W
/020005  /SNAKE 05    / 2/AC:AH1W  /ESC   /010830Z/2830N08200W//
GENTEXT/GENERAL SPINS INFORMATION/THIS IS WHERE GENERAL INFORMATION IS PLACED//
```

Appendix D. TACOPDAT Message Example

The following is an example TACOPDAT message taken from the USMTF Message Browser Help, 2004 Baseline edition [28].

```
EXER/BRAVE SHIELD 93//
MSGID/TACOPDAT/AADC/1221001//
EFFECTIV/12220700Z/2//
8MOVAA
/REF   /ZZPOS      /TIMPOS /CRS/SPD/AAWAX/ANGLE/CMNT
/A001  /3000N07900W/220600Z/330/ 15/  350/  090
/A002  /3030N08000W/221040Z/045/ 15/  350/  090
/A003  /3110N07900W/221500Z/  -/  -/  350/  090//
AAW/CVA-45/ALFA/350-ZZ-50//
AAW/CUSHING/BRAVO/359-ZZ-100//
AAW/CGN1700/CHARLIE/230-ZZ-25//
GND/CRC0700/3320N09920W/351.2/308.1//
GND/CRP0701/3510N10520W/296.2/315.1//
GND/CRP0702/2900N10410W/326.2/286.1//
GND/TAOC0600/3200N09615W/291.2/276.3//
GND/Q730500/3520N10600W/TAD01/TAD02//
GND/Q730501/3200N09206W/TAD03/TAD04//
CAP/STA:JULIETT/ALT:225/-/-/AEW1/270-ZZ-240//
CAP/STA:KILO/ALT:225/-/-/AEW2/350-ZZ-170//
CAP/STA:LIMA/ALT:225/-/-/CRP1/3510N10620W//
CAP/STA:MIKE/ALT:225/-/-/CRP2/2900N10455W/2900N10425W//
CAP/STA:NOVEMBER/ALT:225/-/-/TAOC/3250N09615W//
CAP/STA:OSCAR/ALT:225/-/-/E-3A1/3600N10600W//
CAP/STA:PAPA/ALT:225/-/-/E-3A2/2800N10410W//
CAP/STA:QUEBEC/ALT:225/329.2/341.6/B/359-ZZ-130//
CAP/STA:ROMEO/ALT:225/316.0/249.3/C/220-ZZ-104//
CAP/STA:SIERRA/ALT:H/249.0/315.6/A/280-ZZ-100//
AEW/AEW1/MINALT:180/321.1/283.7/270-ZZ-250//
AEW/AEW2/MINALT:180/302.1/283.7/350-ZZ-190//
AEW/AEW3/MINALT:290/346.0/275.3/3350N09920W/3750N09920W//
AEW/AEW4/MINALT:290/249.0/327.3/3940N10550W/3940N10710W//
AUTOCAT/CINDY/MINALT:205/TX:265.2/REC:328.2/340-ZZ-80//
AAR/STA:DRINK/ALTMN:185/ALTMX:200/289.2/316.4/270-ZZ-160//
AAR/STA:FILLUP/ALTMN:190/ALTMX:205/277.7/333.3/350-ZZ-140//
DUTY/CVA/1//
DUTY/CRC/1S/6//
DUTY/TAOC/1S/6//
SURVEIL/S1/CVA-45/SECLIMIT:240T/SECLIMIT:340T//
SURVEIL/S2/DDG-111/SECLIMIT:340T/SECLIMIT:120T//
SURVEIL/S3/DLG/SECLIMIT:120T/SECLIMIT:240T//
SURVEIL/S4/CRC/-/-/4000N09500W/2600N09500W/2600N12000W/4000N12000W
/4000N09500W//
AMPN/CRC DIVIDES THE AREA AMONG AIR FORCE PARTICIPANTS AND Q-731//
SURVEIL/S5/TAOC/SECLIMIT:035M/SECLIMIT:083M//
AMPN/TAOC HAS AREA WITHIN 200NM OF ITSELF ALONG WITH Q-732//
NARR/DEFENSIVE RESPONSIBILITIES PARALLEL SURVEILLANCE AREAS//
HANDO/CADILLAC/320-ZZ-120//
HANDO/BUICK/010-ZZ-120//
HANDO/OPEL/170-ZZ-120//
HANDO/DODGE/260-ZZ-125//
HANDO/FORD/4000N12000W//
HANDO/CHEVY/2600N12000W//
HANDO/OLDS/3300N10500W//
HANDO/DATSUN/180-CVA-45//
```

```
MEZ/MEZ1/RAD:50NM/350-ZZ-50//
MEZ/MEZ2/RAD:45NM/359-ZZ-100//
MEZ/MEZ3/RAD:45NM/320-ZZ-25//
MEZ/MEZ4/RAD:30NM/3520N10600W//
MEZ/MEZ5/-/3100N09300W/3100N09100W/3300N09100W/3300N09300W
/3100N09300W//
SAFERAD/ALFA/RAD:50NM/CTRBRG:350/WDTH:060/-/UPALT:450/SPD:350
/FROM:120600Z//
SAFERAD/BRAVO/RAD:45NM/CTRBRG:270/WDTH:040/-/UPALT:430/SPD:400
/FROM:120300Z//
SAFERAD/CHARLIE/RAD:45NM/CTRBRG:240/WDTH:040/-/UPALT:440/-
/FROM:120600Z/TO:141200Z//
SAFECOR/SAFE1/WDTH:10NM/MINSPD:110KTSGS/MAXSPD:300KTSGS/LOALT:150
/UPALT:230/-/-/ENTEXTPT:3130N09330W/ENTEXTPT:3130N09445W
/270-CHN 85-40/090-CHN 85-40//
SAFECOR/SAFE2/-/-/-/LOALT:130/UPALT:320/-/-/ENTEXTPT:3130N09000W
/ENTEXTPT:3130N09300W/3130N09100W/3130N09200W//
EMP/EMCON:B/FROM:220600Z/TO:230600Z//
CTRLPT/APPLE/359-CVA-60/50//
MARSH/UNCLE/300-CVA-50/CALLSIGN:PIRATE 01/263.0//
SPDIS/B:32600/170/180-CVA-20//
AKNLDG/YES//
```

Appendix E. STO Message Example

The following is an example STO message taken from the USMTF Message Browser Help, 2004 Baseline edition [28].

```
EXER/GUARDIAN TIGER/97-2//
MSGID/STO/COMAFSPACE/001//
TIMEFRAM/FROM:100001ZOCT1998/TO:102359ZOCT1998//
HEADING/TASKING//
TSKCNTRY/US//
SVCTASK/F//
TASKUNIT/1SWS//
SPACEMSN/WARNING//
TASKSYS/ID:SEON/-/MSN:TECHNICAL INTELLIGENCE//
TASKPER/FROM:101600ZOCT1998/TO:102230ZOCT1998//
AREA/UTR 34321//
GENTEXT/UNIT REMARKS/SUPPORT JTF-SWA OPERATION DURING BURNING WIND,
REF UCCINCCENT MSG DTG 07099Z OCT 1997 FOR VOICE REPORTING
CONSTRAINTS DIRLAUTH APPROVED//
TASKUNIT/2SWS//
SPACEMSN/WARNING//
TASKSYS/ID:DSP/STRAT WARNING/MSNTYP:DIRSUP//
TASKPER/FROM:100001ZOCT1998/TO:102359ZOCT1998//
AREA/ATLANTIC (SLBM)/EAST (ICBM)/PACIFIC (SLBM)//
GENTEXT/TASK SYSTEM REMARKS/DUPLEX FLT 23 AND FLT 34//
TASKUNIT/50SW-4SOPS/NAME:FALCON//
SPACEMSN/SATELLITE C2//
TASKSYS/ID:MILSTAR/-/MSN:UHF COMM FLT1//
TASKPER/FROM:100001ZOCT1998/TO:101300ZOCT1998//
AREA/120W//
GENTEXT/TASK SYSTEM REMARKS/ACTIVATE FLEET BROADCAST ON FLT1 FOR
CONTINUOUS OPS IN SUPPORT OF GUARDIAN TIGER 97-2//
TASKUNIT/50SW-6SOPS/NAME:FALCON//
SPACEMSN/SPACE CONTROL//
TASKSYS/ID:DMSP/-/MSN:WEATHER//
SPACEOBJ/14352/3A//
TASKPER/FROM:101001ZOCT1998/TO:171300ZOCT1998//
GENTEXT/SPACE OBJECT REMARKS/ACTIVATE DMSP F-10 OPERATIONAL LINE
SCANNER FOR 15 MINUTE DURATIONS OVER AOR//
TASKUNIT/30 SW//
SPACEMSN/SPACELIFT//
TASKSYS/ID:TITAN IV/-/MSNTYP:LAUDEP//
MSNSPEC/NATIONAL/10OCT1997/K-18//
GENTEXT/UNIT REMARKS/CINCEUR REQUIRES PAYLOAD FOR STRATEGIC
OPERATIONS ASAP//
TASKUNIT/45 SW//
SPACEMSN/SPACELIFT//
TASKSYS/ID:DELTA II/-/MSNTYP:LAUSUS//
MSNSPEC/GPS IIR2/15OCT1998//
GENTEXT/TASK SYSTEM REMARKS/PREPARE IIR2 FOR IMMEDIATE LAUNCH TO
REPLACE FAILING SVN XX//
TASKUNIT/55SWXS//
SPACEMSN/OTHER//
GENTEXT/UNIT REMARKS/SHUTTLE SUPPORT - ENSURE NO PMI WILL INTERFERE
WITH SUPPORT FROM 0900Z-1200Z 10OCT1997.  PRIORITIZATION UNTIL
FURTHER NOTICE FOR AD HOC WX SUPPORT IS AS FOLLOWS; BOSNIA, KOREA,
SWA//
```

Appendix F. OPTASK LINK Message Example

The following is an example OPTASK LINK message taken from the USMTF Message Browser Help, 2004 Baseline edition [28].

```
OPER/PROVIDENT SWORD//
MSGID/OPTASK LINK/CCG 7/0010/OCT//
POC/JOHN COONTZ/CDR/ABRAHAM LINCOLN/LOC:LINCOLN/TEL:DSN525-1212
/TEL:619-236-2223/EMAIL:COONTZ(AT)LINCOLN.NAVY.MIL//
PERIOD/210001ZMAR/151200ZMAY//
DLRPGRID/DLRP/L5/GPLLM:3600N12600W//
AMPN/DLRP CHANGES WILL BE PROMULGATED VIA DAILY INTENTIONS//
IVCCN/ADCCN/ST800Z1/P/ASGN:4010.5KHZ//
IVCCN/DCN/DT607M1/P/ASGN:3030.0KHZ//
CORRDEC/MAN/2.0/1.0/6/12/3/60/50/20/3/12/1.7/3//
HEADING/MULTILINK INTERFACE COORDINATION REQUIREMENTS//
GENTEXT/REGIONAL INTERFACE INFORMATION/THIS SET IS USED TO PROVIDE
ADDITIONAL GUIDANCE WHEN REGIONAL MULTI-TADIL NETWORKS ARE REQUIRED.
THE JICO WORKS CLOSELY WITH THE RICO IN PLANNING THE REGIONAL
ARCHITECTURES//
GENTEXT/SECTOR INTERFACE INFORMATION/THIS SET IS USED TO PROVIDE
ADDITIONAL GUIDANCE WHEN SECTOR MULTI-TADIL NETWORKS ARE REQUIRED.
THE JICO WORKS CLOSELY WITH THE SICO IN PLANNING THE SECTOR
ARCHITECTURES//
GENTEXT/CHANGE DATA ORDER AUTHORITIES/THIS GENTEXT SET PROVIDES
AMPLIFYING GUIDANCE FOR IDENTIFICATION DIFFERENCE (ID) DIFFERENCE
RESOLUTION PROCEDURES.  THE JICO MAY IN FACT PROMULGATE AN IDENTITY
DIFFERENCE RESOLUTION TABLE THAT MIGHT VARY WITH THE AREA OF
OPERATION//
GENTEXT/COMMAND AUTHORITIES/THIS GENTEXT SET PROVIDES GUIDANCE AND
DIRECTION TO C2 INTERFACE UNITS (IUS) WITH SPECIFIC COMMAND
AUTHORITY.  ONLY THESE C2 UNITS WITH SPECIFIC COMMAND AUTHORITY SHALL
ISSUE INTERFACE COMMANDS.  THE CJTF AND AADC ALWAYS HAVE COMMAND
AUTHORITY.  THE CJTF OR AAD MAY DESIGNATE OTHER C2 IUS AS HAVING THE
AUTHORITY TO ORIGINATE COMMAND WITHIN THE INTERFACE AS NECESSARY.
THIS GUIDANCE WOULD BE PROMULGATE WITHIN THIS SET//
GENTEXT/INTELLIGENCE LOCAL DISCRETE IDENTIFIER/INTELLIGENCE AND ES
(J ONLY) LOCAL DISCRETE IDENTIFIERS (LDIS) ARE DECIMAL NUMBERS FROM
1-4,094 THAT MAY BE DEFINED BY AN OPERATIONAL COMMANDER FOR ANY
DESIRED PURPOSE RELATED TO INTELLIGENCE REPORTING OR EW OPERATIONS.
LDIS ARE NOT THE SAME AS THE PLATFORM DISCRETE IDENTIFIER (DI) CODES
USED BY THE U S NAVY.  THE DI CODES ARE OCTAL NUMBERS AND ARE A FORM
OF SPECIAL CODE//
GENTEXT/CONTINGENCY PROCEDURES/CONTINGENCY PROCEDURES FOR NON-DIGITAL
DATA EXCHANGE.  THE PARTIAL OR COMPLETE LOSS OF THE TADIL INTERFACE
IS ALWAYS A POSSIBILITY.  TO ENSURE THAT OPERATIONS ARE NOT AFFECTED
BY PROLONGED OUTAGES, CONTINGENCY PROCEDURES OUTLINED IN THIS SECTION
WOULD BE OUTLINED FOR EXCHANGING DATA IN A NON-DIGITAL FORM//
INTCOORD/INTERFACE COORDINATION SEGMENT//
AMPN/WHEN THE AREA OF RESPONSIBILITY (AOR) IS DIVIDED AND TWO OR MORE
RICO/SICO(S) ARE DESIGNATED, COMMUNICATIONS CONNECTIVITY BETWEEN EACH
OF THE RICO/SICO(S) AND THE ICO SHOULD BE IDENTIFIED.  THE ICO WILL
SOLICIT INFORMATION FROM EACH DESIGNATED RICO/SICO CONCERNING THE
TECHNICAL PARAMETERS THAT AFFECT THEIR OPERATION//
MULCDUTY/SHIP:LINCOLN/FRANCIS/LCDR/ICO/TEL:619-437-1234
/SECTEL:DSN553-1234/EMAIL:FRANCIS@LINCOLN.NAVY.MIL//
IVCCN/DCN/AB234G/P/TEL:619-437-3456/TELTAC:555-1011//
IVCCN/TSN/AB345A/P/TEL:619-437-6543/TELTAC:555-2314//
MULCDUTY/SHIP:CHOSIN/JOSEFOSKY/LT/RICO/TEL:619-553-8916//
```

IVCCN/DCN/AG342H/P/TEL:619-553-8756/TELTAC:456-2345//
GENTEXT/MULTILINK COORDINATION DETAILS/TO SUPPORT JTF OPERATIONS THE MULTI-TADIL AREA OF OPERATIONS (AO) MAY BE SUB-DIVIDED INTO REGIONS AND SECTORS. IN CASES WHERE REGIONS AND SECTORS ARE EMPLOYED, REGIONAL ICO'S (RICO) AND SECTOR ICO'S (SICO) WILL BE ASSIGNED AND ARE GOVERNED BY THE SAME TECHNIQUES AND PROCEDURES OF THE JICO. GOVERNING COORDINATION INSTRUCTIONS WILL BE PROMULGATED IN THIS SECTION//
GENTEXT/WEB COORDINATION/IN THE PLANNING AND PREPARATION PHASES OF THE OPERATION, THE ICO MAY ELECT TO UTILIZE A WEB PAGE FOR COORDINATION OF A PORTION OF THE OPERATION AND THIS AREA IS DESIGNED TO SUPPORT THOSE REQUIREMENTS//
LNKIV/LINK 4 SEGMENT//
LKIVADDR/UNIT:LINCOLN/ADD:01245//
LKIVADDR/UNIT:CHOSIN/ADD:01237//
LKIVADDR/UNIT:VF-43/BLOCK:02000-02022//
LKFREQC/LINCOLN/DESG1234/ASGN:234.5MHZ//
LKFREQC/CHOSIN/DESG2345/ASGN:245.6MHZ//
LNKXI/LINK 11 SEGMENT//
LSYSDATA/13-9/ON/A2/SLEW//
CRYPTDAT/AKAI1238/AMAS23/KG-40/AMASL6789/2345Z//
DALKFREQ/DATA/FD:DATA01/ASGN:3037.5KHZ/P/B7D/D//
DALKFREQ/DATA/FD:AC11/ASGN:2.2440MHZ/P/B7D//
DALKFREQ/DATA/FD:AC11/ASGN:9.5770MHZ/S/B7D//
DALKFREQ/DATA/FD:AC11A/ASGN:398.3MHZ/P/F2D//
DALKFREQ/DATA/FD:AC11A/ASGN:395.4MHZ/S/F2D//
FORCFLTR/M1/ALL/-/-/-/SR//
FORCFLTR/M2/ALL/-/-/-/AL//
LPUDATA/SHIP:LINCOLN/CS:L/PU:66/BLOCK:1000-1376/-/83/84/85/88/804//
LPUDATA/SHIP:CHOSIN/CS:Q/PU:65/BLOCK:2000-2776/-/81/83S/84S/806//
LPUDATA/SHIP:BUNKER HILL/CS:K/PU:23/BLOCK:3000-3576/-/83S/84S//
LPUDATA/SHIP:JONES/CS:B/PU:53/BLOCK:4000-4776//
LNKXIB/LINK 11B SEGMENT//
CRYPTDAT/AKAK1238/AMAS23/KG-84/AMASL6789/2345Z//
LRULINK/RUNO:140/RUNO:145/2400/PRI/LTD//
LRULINK/RUNO:150/RUNO:155/2400/PRI/LTD//
LRULINK/RUNO:101/RUNO:105/2400/ALT/FTD//
LRULINK/RUNO:130/RUNO:136/2400/ALT/FTD//
DALKFREQ/DATA/FD:DATA01/ASGN:3037.5KHZ/P/B7D/D//
FORCFLTR/FAB28/ALL/-/STATIC/-/SR//
LRUDATA/UNIT:1-1 BN/CS:FANG/RU:155/4400-5300//
LRUDATA/UNIT:1-2 BN/CS:CLAW/RU:146/5500-5600//
LRUDATA/UNIT:1-3 BN/CS:BLACKCREEK/RU:125/5600-5700//
UNITFLTR/A14/ALL/-/STATIC/-/SR//
NARR/ALL ALTERNATE LINKS WILL BE ACTIVATED BY ICO//
LNKXVI/LINK 16 SEGMENT//
JNETWORK/ACDO0002A/232/09DEC2001/017/PRI//
CPD/12SEP2002/1//
JCRYPDAT/1/USKAT-9017/-/0/-/1//
JTRNMODE/TEST2/OFF/NOR/MODE 1//
JSTNETS/CNTRL/19/123/DECATUR//
JSTNETS/SFTF/19/039/VF31//
FORCFLTR/FAB28/ALL/9/STATIC/-/SR//
FORCFLTR/M1/ALL/-/SLAVED/APD:XX/AL//
GENTEXT/FORCE FILTER SUMMARY/FILTER M1 IS THE RECTANGLE MARSHALL AREA OF THE CARRIER. DURING CVN RECOVERY, ALL UNITS (IF CAPABLE) WILL TRANSMIT FILTER ALL FRIENDLY TRACK AROUND CVN (INCLUDING MARSHALL). FILTER SHOULD BE SET AS FOLLOWS DEPENDING ON CASE RECOVERY: CASE I, XY COORD FILTER CENTERED ON CVN/20NM AHEAD AND 20NM RECIPROCAL OF BASE RECOVERY COURSE/15NM ON EITHER SIDE OF CVN. CASE II/III, XY COORD FILTER CENTERED ON CVN/10NM AHEAD AND 35 NM (OR REQUIRED DISTANCE TO INCLUDE MARSHALL) ON RECIPROCAL OF BASE RECOVERY COURSE/ 15NM ON EITHER SIDE OF CVN. UNITS UNABLE TO FILTER XY AND OPERATING

```
IN CLOSE PROXIMITY TO CVN (WITHIN 75NM) WILL SWITCH TO RCV-ONLY OR
TRANSMIT INHIBIT TRACKS IN VICINITY OF THE CVN DURING LAUNCH/RECOVERY
CYCLES//
JUDATA/SHIP:LINCOLN/-/-/PRI:00062/16/BLOCK:00500-00700/-/SHIP(4)/12
/NORM/0200/PRI/Y//
JCNTROPT/2/1//
LKSXDUTY/820/821/823//
JUDATA/SHIP:CHOSIN/-/-/PRI:00023/16/BLOCK:00300-00400/-/SHIP(5)/13
/NORM/0200/PRI/Y//
JCNTROPT/2/2//
LKSXDUTY/823//
JUDATA/SHIP:BUNKER HILL/-/-/PRI:00042/16/BLOCK:00100-00200/-/SHIP(6)
/14/NORM/0200/PRI/Y//
JCNTROPT/2/3//
LKSXDUTY/825S/823S//
JUDATA/SHIP:JONES/-/-/PRI:00022/16/BLOCK:00200-00300/-/SHIP(7)/15
/NORM/0200/PRI/Y//
JCNTROPT/2/4//
LKSXDUTY/821S//
LNKSAT/SATELLITE LINK 16 SEGMENT//
DAMA/34256/34/4800BPS//
SATINFOJ/23/CHAN45/345.78MHZ/34//
CRYPSATJ/KG-84/AMALS6789//
LNKJPADD/SHIP:LINCOLN/-/PU:62/BLOCK:0300-0400/C2P/YES/NO//
SPECTRK/LXVI:AB345/SAM SITE CHARLIE//
SPECTRK/LXVI:AB456/RADAR SITE ALPHA//
```

Appendix G. Sample GAMS Model Code for Polymorphic Networking

This is an example copy of the GAMS file whose model, when solved, generates a polymorphic network. The characteristics of the network are placed into three separate files that get included into the master file (polynet3.gms) at compile time. The listing below shows the results of the inclusion. In this case the network has three nodes, with two interface types, and three commodities (out of a maximum of six). Such a small network has been chosen for an example to keep the length of the listing to a reasonable size. The values in the tables are for the most part randomly generated at when the include files are generated. Tables that get filled in after solutions are found are initially set to all zeroes. Lines that begin with '*' and all text after '!!' are comments. Texts within quotation marks are descriptions of objects. The command line GAMS call is:

```
gams polynet3.gms --MyFile=main_3_3_2 --MyOutFile=3_3_2_0.data pw=170 o=3_3_2_0.lst nodlim=20000000 mip=COINCBC
```

Here '--MyFile=main_3_3_2' sets the variable naming the main included file. The other two included files are referenced within the main_3_3_2 file. The '_3_3_2' in the file name indicates the number of nodes, commodities, and interface types for the network. The argument '--MyOutFile=3_3_2_0.data' sets the variable for the filename to which specialized output is saved, as specified in the model code. The '0' in the filename indicates that this is case 0 in a set of multiple runs. The argument 'pw=170' sets the page width of the listing file to 170 characters. The argument 'o=3_3_2_0.lst' sets the filename for the GAMS listing file. The argument 'nodlim=20000000' sets the node limit for mixed integer programming. Finally, the argument 'mip=COINCBC' tells GAMS to use COINCBC as the solver. The model code now follows.

```
$title “Polymorphic networking”         !! Sets title in page header of the listing file
$oneolcom oninline offinclude onglobal  !! Turn on eol and inline comments, turn off listing of
                                        !! include file names, force inheritance of parent file settings
* $offlisting                           !! prevents echoing of input file into output file
* $onsymlist                            !! turn on symbol listing
* $onsymxref                            !! turn on symbol cross reference listing
* $onuellist                            !! turn on unique element listing
* $onuelxref                            !! turn on unique element cross reference

File MyFile    "Input file for network characteristics";  !! To be assigned in command line gams call
$if not set MyFile $abort “>>> TOO FEW ARGUMENTS <<<”
File MyOutFile "Output file for network characteristics"; !! To be assigned in command line gams call
$if not set MyOutFile $abort “”>>> TOO FEW ARGUMENTS <<<”
Put MyOutFile;                                            !! Assigns MyOutFile as current file written to
MyOutFile.nw = 0;                                         !! Sets the numeric field width to be variable
MyOutFile.lw = 0;                                         !! Sets the label field width to be variable
MyOutFile.nd = 4;                                         !! Sets the number of decimals to 4

Scalar NumNodes       "The number of nodes in the network" /3/
       NumInterfaces  "The number of interface types"      /2/
       NumCommodities "The number of commodities"          /3/

Sets TableHeader1 "Headers for SourceDest"  /"source", "dest", "bandwidth"/
     TableHeader2 "Header for Hops"         /"hops"/
     Nodes        "Set of node labels"      /1*3/     !! 1 through NumNodes
     Interfaces   "Set of interfaces"       /1*2/     !! 1 through NumInterfaces
     Commodities  "Set of commodity labels" /1*3/;    !! 1 through NumCommodities

Alias (Nodes, Nodes2);                                  !! Allows one to doubly index over set of nodes to form edges

Table NodeInterfaces(Nodes, Interfaces) "Table listing the number of each type of interface at each node"
       1       2
1      4       2
2      2       2
3      2       4;

Table SourceDest(Commodities, TableHeader1) "Table of source, destination, and required bandwidth for each commodity"
       source  dest    bandwidth
1      1       2       4
2      2       1       12
3      1       3       7;

Table r(Nodes, Commodities) "Used to help define ConserveFlow equation"
       1       2       3
1      1       -1      1
2      -1      1       0
3      0       0       -1;

Table Hops(Commodities, TableHeader2) "Table of number of hops for each commodity" !! Filled in after solution found
       hops
1      0
2      0
3      0;

Table FCost(Nodes, Nodes2, Interfaces) "Table of fixed costs for each directed edge (i,j,f)"
       1       2
1.2    9       9
1.3    9       10
2.1    8       8
2.3    5       8
3.1    5       5
3.2    10      10;

Table A(Nodes, Nodes2, Interfaces) "Table of arc possibilities (node incidence matrix)"
       1       2
1.2    1       1
1.3    0       1
2.1    1       1
2.3    1       1
3.1    0       1
3.2    1       1;

Table Yold(Nodes, Nodes2, Interfaces) "Whether or not edge (i,j,f) was included in previous topology"
       1       2
1.2    0       0
1.3    0       0
2.1    0       0
2.3    0       0
3.1    0       0
3.2    0       0;

Table Cap(Nodes, Nodes2, Interfaces) "Table of arc capacities"
       1       2
1.2    3       5
1.3    3       4
2.1    6       6
2.3    5       4
3.1    3       5
3.2    6       4;
```

```
Table VarCost(Nodes, Nodes2, Interfaces, Commodities) "Table giving cost for 100% of commodity j to flow on (i, j, f)"
        1       2       3
1.2.1   9       9       12
1.2.2   7       15      8
1.3.1   15      7       14
1.3.2   13      13      12
2.1.1   13      15      10
2.1.2   8       10      13
2.3.1   15      14      8
2.3.2   7       7       11
3.1.1   7       9       7
3.1.2   8       9       14
3.2.1   10      11      13
3.2.2   7       8       9;

Table AddCost(Nodes, Nodes2, Interfaces, Commodities) "Table giving additional cost for 100% of commodity j to flow on
(i, j, f)"
        1       2       3
1.2.1   0       0       0
1.2.2   0       0       0
1.3.1   0       0       0
1.3.2   0       0       0
2.1.1   0       0       0
2.1.2   0       0       0
2.3.1   0       0       0
2.3.2   0       0       0
3.1.1   0       0       0
3.1.2   0       0       0
3.2.1   0       0       0
3.2.2   0       0       0;

Table Xold(Nodes, Nodes2, Interfaces, Commodities) "Previous percentage of commodity k to flow on (i, j, f)"
        1       2       3
1.2.1   0       0       0
1.2.2   0       0       0
1.3.1   0       0       0
1.3.2   0       0       0
2.1.1   0       0       0
2.1.2   0       0       0
2.3.1   0       0       0
2.3.2   0       0       0
3.1.1   0       0       0
3.1.2   0       0       0
3.2.1   0       0       0
3.2.2   0       0       0;

set Sold(Commodities) "Whether or not commodity k was dropped in previous topology"
/1      0
 2      0
 3      0/;

Scalar Diameter           "Maximum number of hops among all commodities"                          /0/
       AvgNumHops         "Average number of hops per commodity"                                  /0/
       DroppedCommodities "How many commodities were dropped to achieve feasibility"              /0/
       Difference         "Measurement of difference between current solution and previous soloution" /0/
       Timer              "Keeps track of how long solve times take"
       DeltaTime          "Actual elapsed time for a solve";

Variables tot  "Total cost of designing and routing network"
          fix  "Fixed cost of designing network"
          var  "Variable cost for routing choice"
          addl "Additional cost for routing choice";

Positive Variables X(Nodes, Nodes2, Interfaces, Commodities) "Percentage of commodity k to flow on (i,j,f)";
X.up(Nodes, Nodes2, Interfaces, Commodities) = A(Nodes, Nodes2, Interfaces);

Binary Variables Y(Nodes, Nodes2, Interfaces) "Decision variable for whether edge (i,j,f) is included in topology"
                 S(Commodities)               "Decision variable for whether commodity k should be dropped";

Equations           !! Declare equations
    TotalCost                       "Total cost of the network" !! Objective function
    FixedCost                       "Construction cost"         !! Cost from including each edge
    VariableCost                    "Routing cost"              !! Cost from routing commodities
    AdditionalCost                  "Additional routing cost"   !! Penalty cost to inhibit repeating topologies
    ConserveFlow(Nodes, Commodities) "Flow balance equations"    !! Flow in = Flow out unless source or destination
    LinkCapacity(Nodes, Nodes2, Interfaces) "Link capacity constraints"   !! Only send flow that an edge can handle
    Degree(Nodes, Interfaces)               "Interface constraints"       !! Don't exceed a node's number of interfaces
    Forcing(Nodes, Nodes2, Interfaces, Commodities) "Forcing constraints" !! X(i,j,f,k) <= Y(i,j,f)
    Inclusion(Nodes, Nodes2, Interfaces)    "Inclusion constraints"       !! Y(i,j,f) <= A(i,j,f)
    Bidirection(Nodes, Nodes2, Interfaces)  "Bidirectional constraints";  !! if (i,j,f) exists, then so should (j,i,f)

!! Define equations
TotalCost .. tot =e= fix + var + addl + sum(Commodities, 1000*SourceDest(Commodities, "bandwidth")*S(Commodities));
FixedCost .. fix =e= sum((Nodes, Nodes2, Interfaces), Y(Nodes, Nodes2, Interfaces)*FCost(Nodes, Nodes2, Interfaces));
VariableCost .. var =e= sum((Nodes, Nodes2, Interfaces, Commodities), X(Nodes, Nodes2, Interfaces, Commodities)* _
    VarCost(Nodes, Nodes2, Interfaces, Commodities));
AdditionalCost .. addl =e= sum((Nodes, Nodes2, Interfaces, Commodities), X(Nodes, Nodes2, Interfaces, Commodities)* _
    AddCost(Nodes, Nodes2, Interfaces, Commodities));
ConserveFlow(Nodes, Commodities) .. sum((Nodes2, Interfaces),X(Nodes, Nodes2, Interfaces, Commodities))- _
    sum((Nodes2, Interfaces),X(Nodes2, Nodes, Interfaces, Commodities)) =e= r(Nodes, Commodities)*(1-S(Commodities));
LinkCapacity(Nodes, Nodes2, Interfaces) .. sum(Commodities, X(Nodes, Nodes2, Interfaces, Commodities)* _
    SourceDest(Commodities, "bandwidth")) =l= Cap(Nodes, Nodes2, Interfaces);
```

```
Degree(Nodes, Interfaces) .. sum(Nodes2, Y(Nodes, Nodes2, Interfaces)) =l= NodeInterfaces(Nodes, Interfaces);
Forcing(Nodes, Nodes2, Interfaces, Commodities) .. X(Nodes, Nodes2, Interfaces, Commodities) =l= _
    Y(Nodes, Nodes2, Interfaces);
Inclusion(Nodes, Nodes2, Interfaces)           .. Y(Nodes, Nodes2, Interfaces) =l= A(Nodes, Nodes2, Interfaces);
Bidirection(Nodes, Nodes2, Interfaces)         .. Y(Nodes, Nodes2, Interfaces) =e= Y(Nodes2, Nodes, Interfaces);

!! Define model and set options
Model net /all/;
net.optcr = 0;                                 !! relative termination criteria for MIP
Option limrow = 0;                             !! Larger values let you see more results in the output file
Option limcol = 0;                             !! Larger values let you see more results in the output file
Option solprint = off;                         !! Turning on lets you see more results in the output file
Option iterlim = 2000000000;                   !! Number of solver iterations (as high as possible)
Option reslim = 100000000;                     !! Amount of solver time in seconds (as high as possible)

Put "# Solving initial network", ": Elapsed time ";

!! Solve model while timing
Timer = timeelapsed;
Solve net using mip minimizing tot;            !!***********SOLVE STATEMENT HERE!!!
DeltaTime = timeelapsed - timer;
Put DeltaTime /;

Loop((Nodes, Nodes2, Interfaces, Commodities)$(X.l(Nodes, Nodes2, Interfaces, Commodities)>0),
    Put Nodes.tl "." Nodes2.tl "." Interfaces.tl "." Commodities.tl;
    Put , "  ", X.l(Nodes, Nodes2, Interfaces, Commodities) /;
);

Loop((Nodes, Nodes2, Interfaces, Commodities)$(X.l(Nodes, Nodes2, Interfaces, Commodities) > 0),
    Hops(Commodities, "hops") = Hops(Commodities, "hops") + 1;
    AddCost(Nodes, Nodes2, Interfaces, Commodities) = AddCost(Nodes, Nodes2, Interfaces, Commodities) + _
        X.l(Nodes, Nodes2, Interfaces, Commodities);
);

!! Count the number of dropped commodities
Loop((Commodities)$(S.l(Commodities) > 0), DroppedCommodities = DroppedCommodities + 1;);

Diameter = smax(Commodities, Hops(Commodities, "hops"));
AvgNumHops = sum(Commodities, Hops(Commodities, "hops"))/(NumCommodities-DroppedCommodities);

Put "Total cost: ", tot.l /;
Put "True cost: ", (tot.l - addl.l) /;
Display X.l, Y.l, S.l, Hops, Diameter, AvgNumHops, DroppedCommodities, fix.l, var.l, addl.l, tot.l;

!! Prepare for the next iteration
DroppedCommodities = 0;
Loop((Commodities),
    Hops(Commodities, "hops") = 0;
    Sold(Commodities) = S.l(Commodities);
    Loop((Nodes, Nodes2, Interfaces),
        Yold(Nodes, Nodes2, Interfaces) = Y.l(Nodes, Nodes2, Interfaces);
        Xold(Nodes, Nodes2, Interfaces, Commodities) = X.l(Nodes, Nodes2, Interfaces, Commodities);
    );
);

Model polynet1 /all/;          !! Define model
polynet1.optcr = 0;            !! relative termination criteria for MIP

Scalar count; count = 1;       !! declare and initialize loop control variable
Scalar limit; limit = 9;       !! declare and initialize loop limit variable
While((count le limit),
    MyOutFile.nd = 0;
    Put /, "# Solving polynet ", count, " of ", limit, ": Elapsed time ";
    MyOutFile.nd = 4;

    !! Solve model while timing
    Timer = timeelapsed;
    Solve polynet1 using mip minimizing tot;       !!***********SOLVE STATEMENT HERE!!!
    DeltaTime = timeelapsed - timer;
    Put DeltaTime /;

    Loop((Nodes, Nodes2, Interfaces, Commodities)$(X.l(Nodes, Nodes2, Interfaces, Commodities)>0),
        Put Nodes.tl "." Nodes2.tl "." Interfaces.tl "." Commodities.tl;
        Put , "  ", X.l(Nodes, Nodes2, Interfaces, Commodities) /;
    );

    !! Count number of hops for each commodity, diameter, and average number of hops and increase AddCost
    Loop((Nodes, Nodes2, Interfaces, Commodities)$(X.l(Nodes, Nodes2, Interfaces, Commodities) > 0),
        Hops(Commodities, "hops") = Hops(Commodities, "hops") + 1;
        AddCost(Nodes, Nodes2, Interfaces, Commodities) = AddCost(Nodes, Nodes2, Interfaces, Commodities) + _
            X.l(Nodes, Nodes2, Interfaces, Commodities);
    );

    !! Count the number of dropped commodities
    Loop((Commodities)$(S.l(Commodities) > 0), DroppedCommodities = DroppedCommodities + 1;);

    Diameter = smax(Commodities, Hops(Commodities, "hops"));
    AvgNumHops = sum(Commodities, Hops(Commodities, "hops"))/(NumCommodities-DroppedCommodities);
    Difference = sum((Commodities),SourceDest(Commodities, "bandwidth")*sum((Nodes, Nodes2, Interfaces), _
        abs(X.l(Nodes, Nodes2, Interfaces, Commodities)-Xold(Nodes, Nodes2, Interfaces, Commodities))))/ _
        (sum((Commodities),SourceDest(Commodities, "bandwidth"))*sum((Nodes, Nodes2, Interfaces), _
        (Y.l(Nodes, Nodes2, Interfaces)+Yold(Nodes, Nodes2, Interfaces))/2));
```

```
    Put "Total cost: ", tot.l /;
    Put "True cost: ", (tot.l - addl.l) /;
    Put "Difference ", Difference /;
    Display X.l, Xold, Y.l, Yold, S.l, Sold, Hops, Diameter, AvgNumHops, DroppedCommodities, Difference;
    Display fix.l, var.l, addl.l, tot.l;

    !! Prepare for the next iteration
    DroppedCommodities = 0;
    Loop((Commodities),
        Hops(Commodities, "hops") = 0;
        Sold(Commodities) = S.l(Commodities);
        Loop((Nodes, Nodes2, Interfaces),
            Yold(Nodes, Nodes2, Interfaces) = Y.l(Nodes, Nodes2, Interfaces);
            Xold(Nodes, Nodes2, Interfaces, Commodities) = X.l(Nodes, Nodes2, Interfaces, Commodities);
        );
    );

    count = count + 1;
);

putclose MyOutFile;     !! Close MyOutFile output file prior to ending program
```

Appendix H. Sample Polymorphic Network

The following data shows the results of a sample GAMS run to create a polymorphic network consisting of five nodes, each with four interfaces. Every conceivable edge is allowed. There are five commodities, one for each node. The destination for the commodity from node n is node $(n+1)\ mod\ 5$. All fixed and variable costs are set to 1. The capacity of every edge is 100 Kbps and each commodity has a required bandwidth of 10 Kbps. An initial (optimal) network topology is found followed by nine polymorphisms. The results shown here come from the MyOutFile mentioned in Appendix G and list every edge that has traffic flowing on it. The format is '(from node).(to node).(interface).(commodity) (% of flow)'. The time to solve is listed along with the total cost and true cost of the network. Note that solution times vary from 20.4430 seconds to 295.1990 seconds. Measured differences range from 0.2000 to 0.4000. True costs range from the optimal 15.0000 of the first topology to 16.0000 for 6 of the 9 subsequent topologies.

```
# Solving initial network: Elapsed time 60.2210
1.2.2.1  1.0000
2.3.4.2  1.0000
3.4.3.3  1.0000
4.5.1.4  1.0000
5.1.4.5  1.0000
Total cost: 15.0000
True cost: 15.0000

# Solving polynet 1 of 10: Elapsed time 46.0010
1.2.3.1  1.0000
2.3.2.2  1.0000
3.4.1.3  1.0000
4.5.3.4  1.0000
5.1.2.5  1.0000
Total cost: 15.0000
True cost: 15.0000
Difference 0.2000
```

```
# Solving polynet 2 of 10: Elapsed time 28.1230
1.2.4.1  1.0000
2.3.1.2  1.0000
3.4.4.3  1.0000
4.5.2.4  1.0000
5.1.1.5  1.0000
Total cost: 15.0000
True cost: 15.0000
Difference 0.2000

# Solving polynet 3 of 10: Elapsed time 23.8390
1.2.1.1  1.0000
2.3.3.2  1.0000
3.4.2.3  1.0000
4.5.4.4  1.0000
5.1.3.5  1.0000
Total cost: 15.0000
True cost: 15.0000
Difference 0.2000

# Solving polynet 4 of 10: Elapsed time 295.1990
1.3.1.1  1.0000
2.3.2.2  1.0000
3.1.1.5  1.0000
3.2.2.1  1.0000
3.5.4.3  1.0000
4.5.2.4  1.0000
5.3.4.5  1.0000
5.4.2.3  1.0000
Total cost: 18.0000
True cost: 16.0000
Difference 0.2889

# Solving polynet 5 of 10: Elapsed time 242.0370
1.3.3.1  1.0000
1.4.4.3  1.0000
2.3.4.2  1.0000
3.1.3.3  1.0000
3.2.4.1  1.0000
4.1.4.5  1.0000
4.5.1.4  1.0000
5.4.1.5  1.0000
Total cost: 18.0000
True cost: 16.0000
Difference 0.4000
```

```
# Solving polynet 6 of 10: Elapsed time 171.6240
1.2.2.1  1.0000
2.1.2.5  1.0000
2.5.1.2  1.0000
3.4.2.3  1.0000
3.5.3.4  1.0000
4.3.2.4  1.0000
5.2.1.5  1.0000
5.3.3.2  1.0000
Total cost: 18.0000
True cost: 16.0000
Difference 0.4000

# Solving polynet 7 of 10: Elapsed time 20.4430
1.4.2.1  1.0000
2.3.3.2  1.0000
2.4.4.3  1.0000
3.2.3.3  1.0000
4.1.2.5  1.0000
4.2.4.1  1.0000
4.5.3.4  1.0000
5.4.3.5  1.0000
Total cost: 18.0000
True cost: 16.0000
Difference 0.4000

# Solving polynet 8 of 10: Elapsed time 97.8190
1.2.4.1  1.0000
2.1.4.5  1.0000
2.4.1.2  1.0000
2.5.2.4  1.0000
3.4.3.3  1.0000
4.2.1.4  1.0000
4.3.3.2  1.0000
5.2.2.5  1.0000
Total cost: 18.0000
True cost: 16.0000
Difference 0.4000

# Solving polynet 9 of 10: Elapsed time 68.3950
1.4.4.1  1.0000
1.5.2.4  1.0000
2.3.1.2  1.0000
2.4.2.3  1.0000
3.2.1.3  1.0000
4.1.4.4  1.0000
4.2.2.1  1.0000
5.1.2.5  1.0000
Total cost: 18.0000
True cost: 16.0000
Difference 0.4000
```

Appendix I. Interval Plots for Scenario 1

The following interval plots show the 95% confidence intervals (CI) for the mean percentages of packets dropped at the router for Scenario 1 (no NTO). Each plot shows the 95% CI for packets from a single source (S1or S2) of a fixed size broken down by the sizes for the packets that come from the opposing source (S2 or S1, respectively).

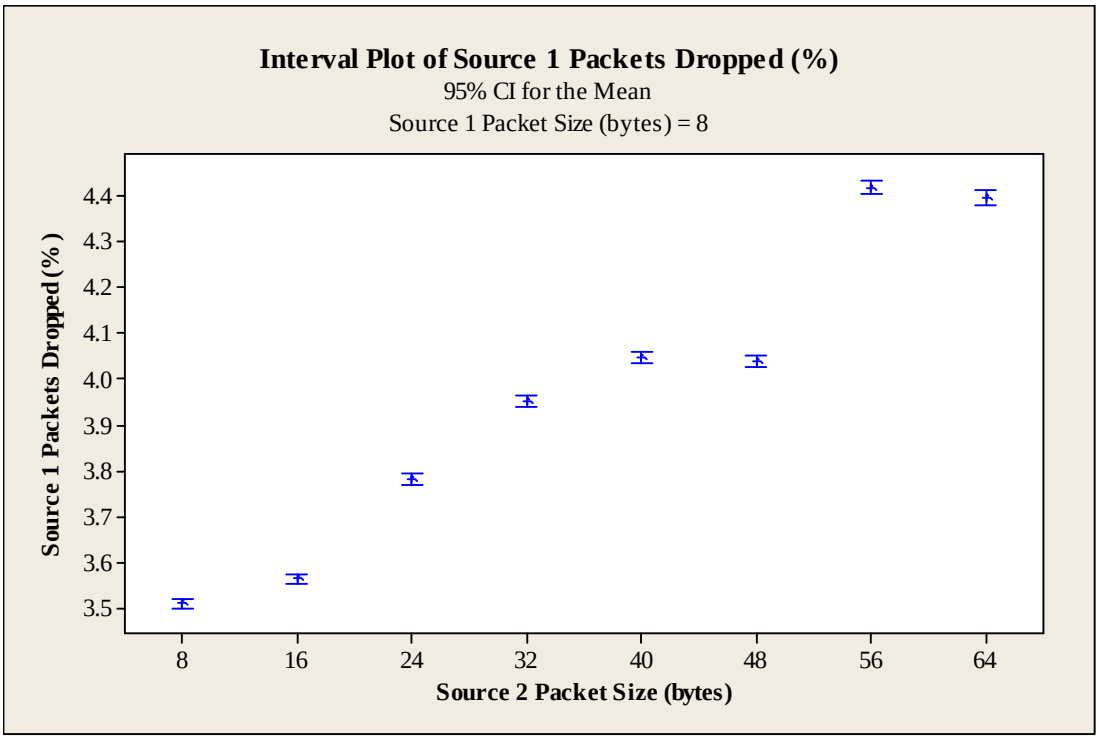

Figure 45: 95% CI for mean % of 8-B S1 packets dropped in Scenario 1 (no NTO)

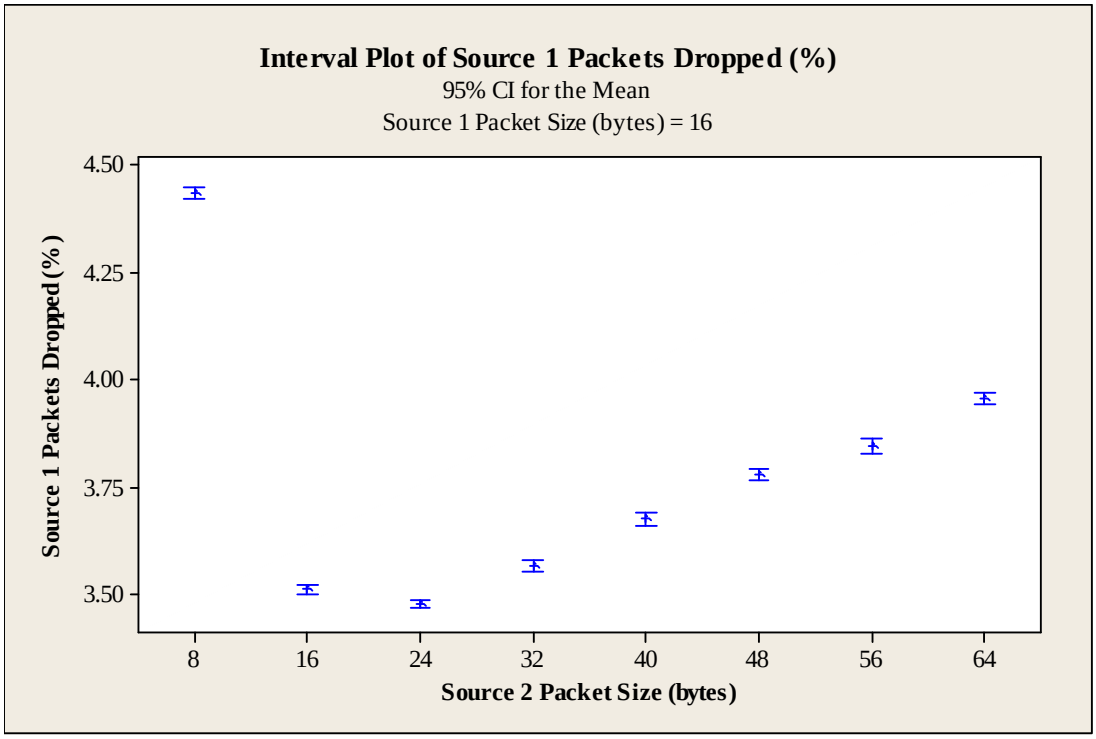

Figure 46: 95% CI for mean % of 16-B S1 packets dropped in Scenario 1 (no NTO)

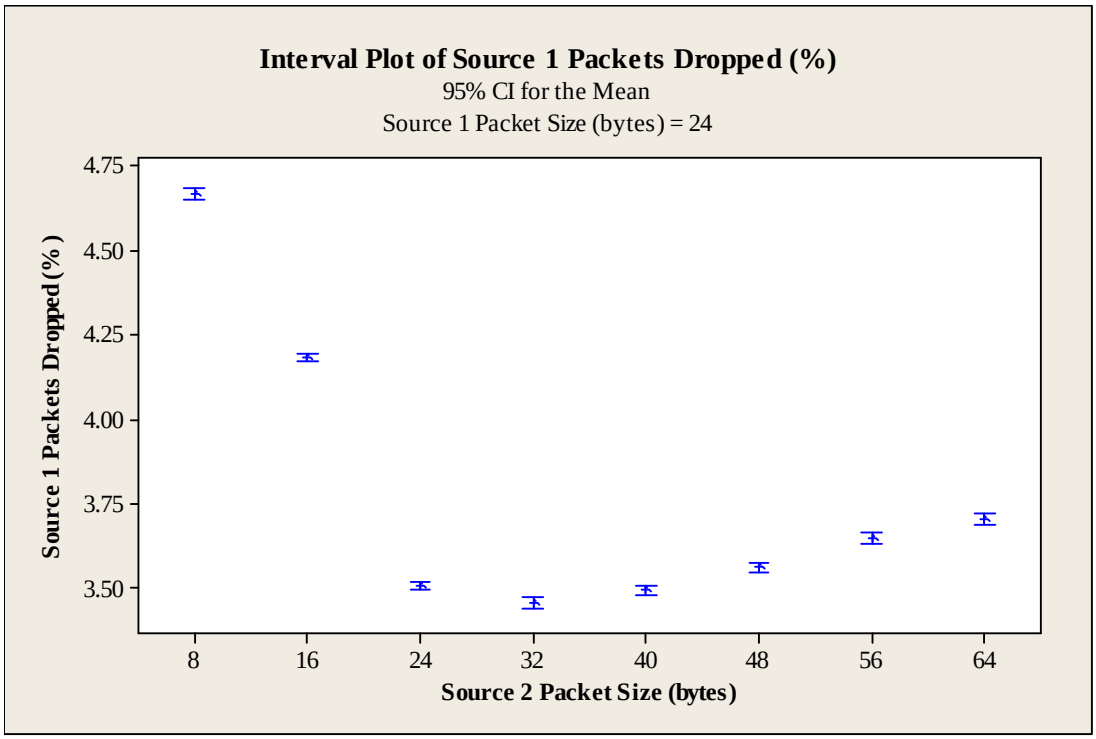

Figure 47: 95% CI for mean % of 24-B S1 packets dropped in Scenario 1 (no NTO)

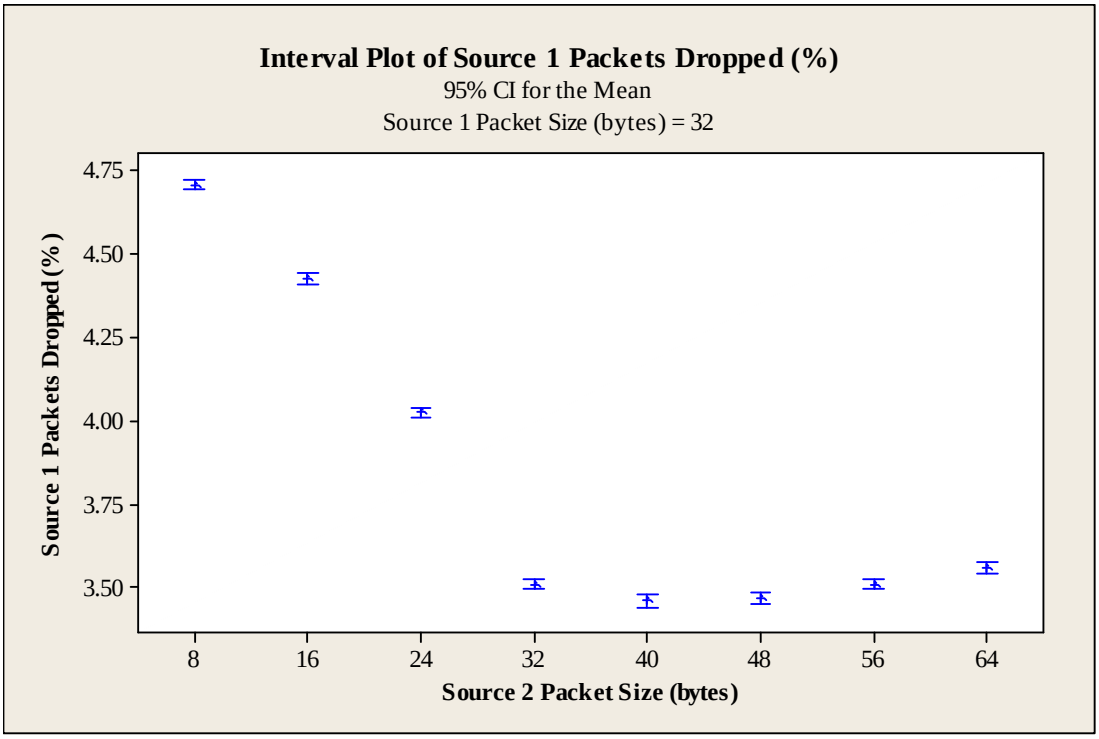

Figure 48: 95% CI for mean % of 32-B S1 packets dropped in Scenario 1 (no NTO)

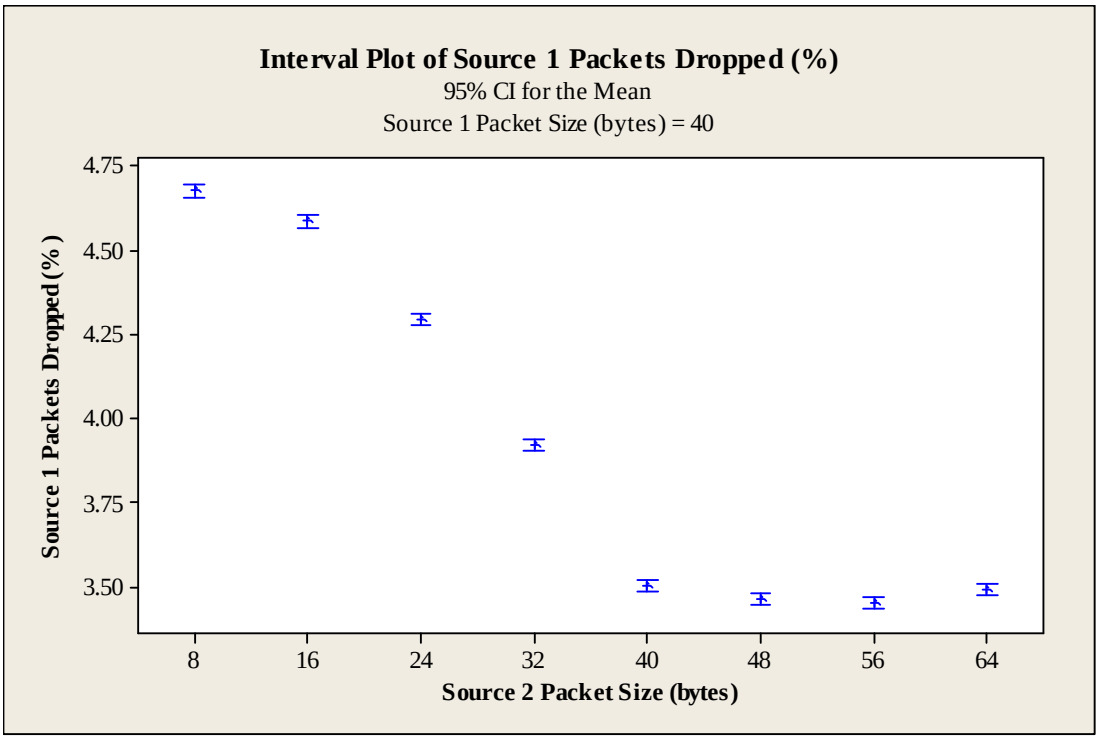

Figure 49: 95% CI for mean % of 40-B S1 packets dropped in Scenario 1 (no NTO)

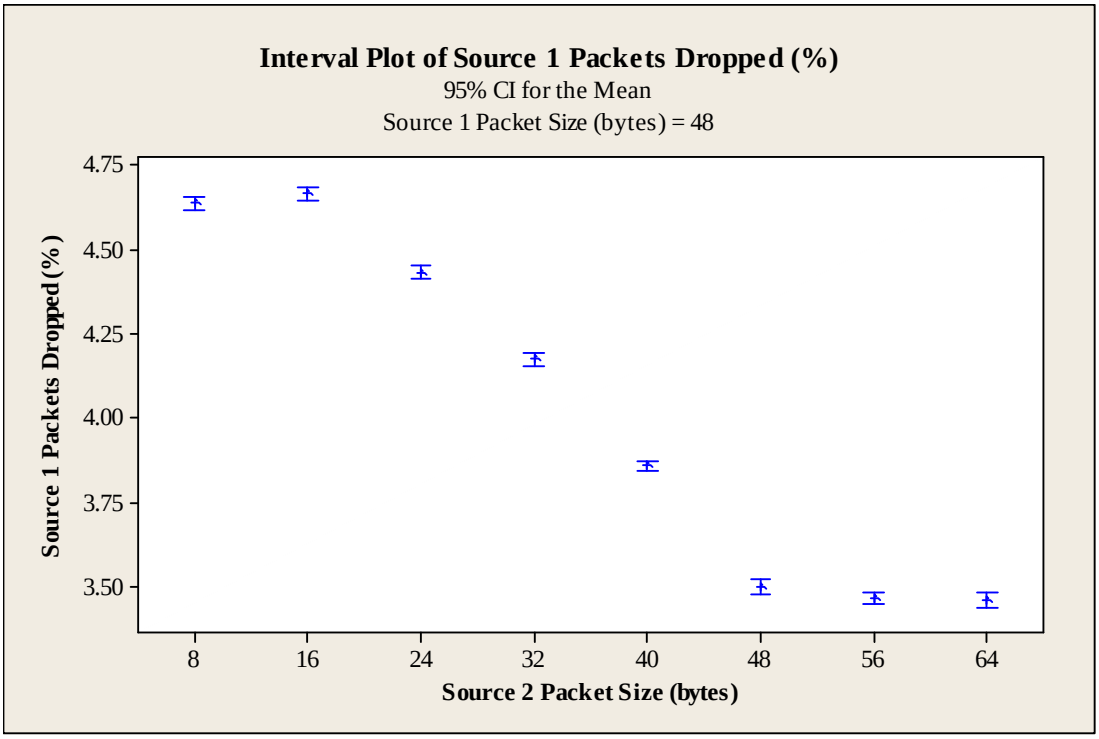

Figure 50: 95% CI for mean % of 48-B S1 packets dropped in Scenario 1 (no NTO)

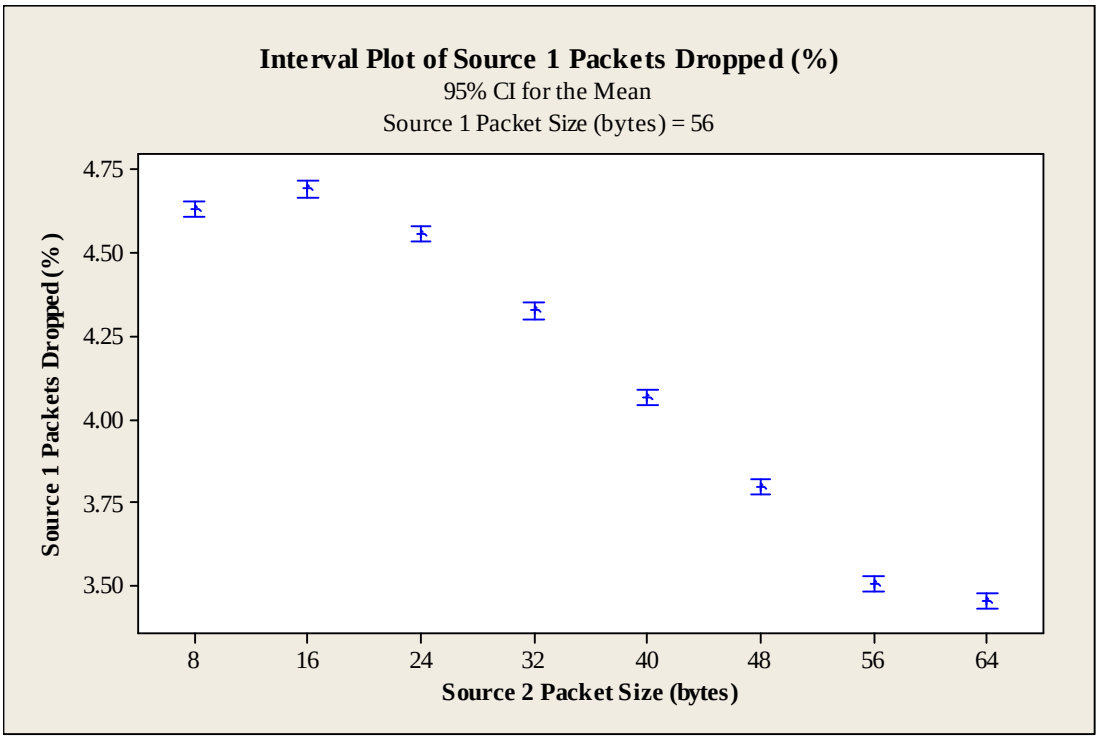

Figure 51: 95% CI for mean % of 56-B S1 packets dropped in Scenario 1 (no NTO)

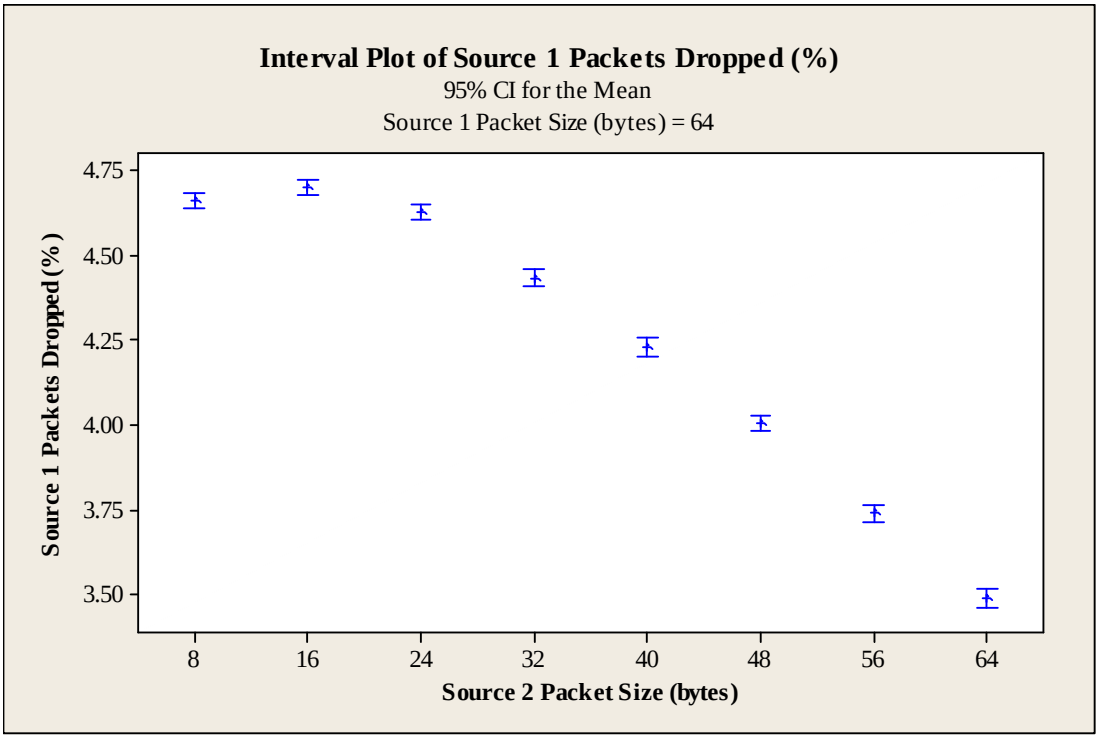

Figure 52: 95% CI for mean % of 64-B S1 packets dropped in Scenario 1 (no NTO)

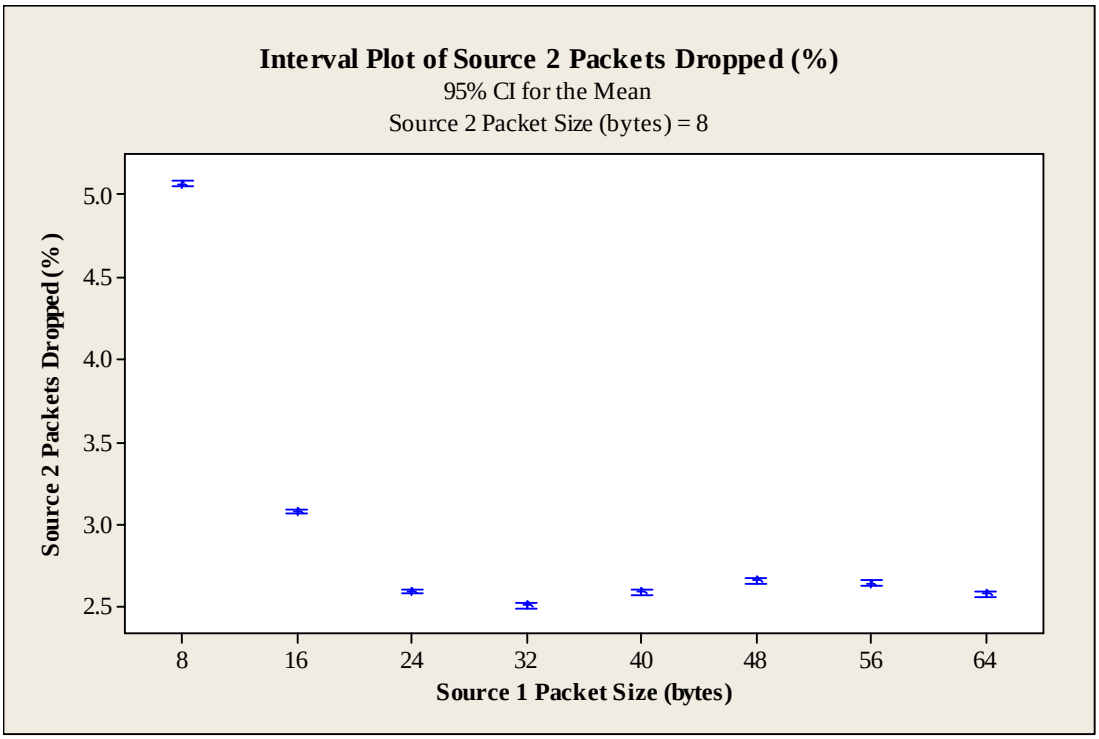

Figure 53: 95% CI for mean % of 8-B S2 packets dropped in Scenario 1 (no NTO)

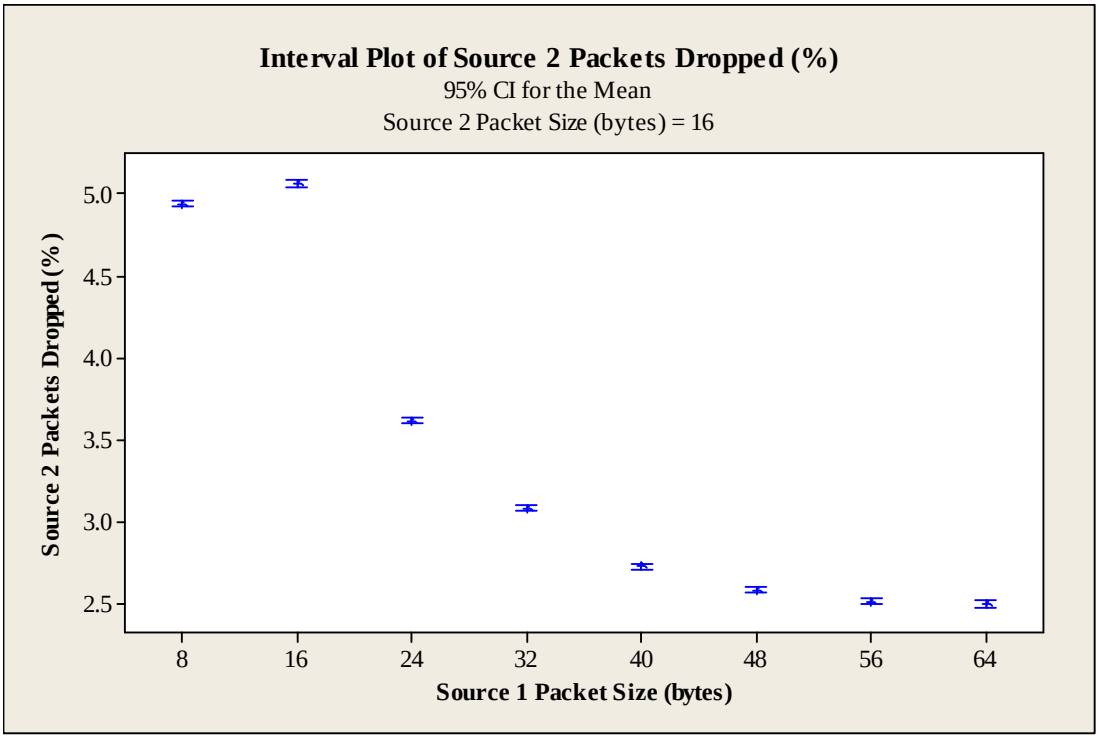

Figure 54: 95% CI for mean % of 16-B S2 packets dropped in Scenario 1 (no NTO)

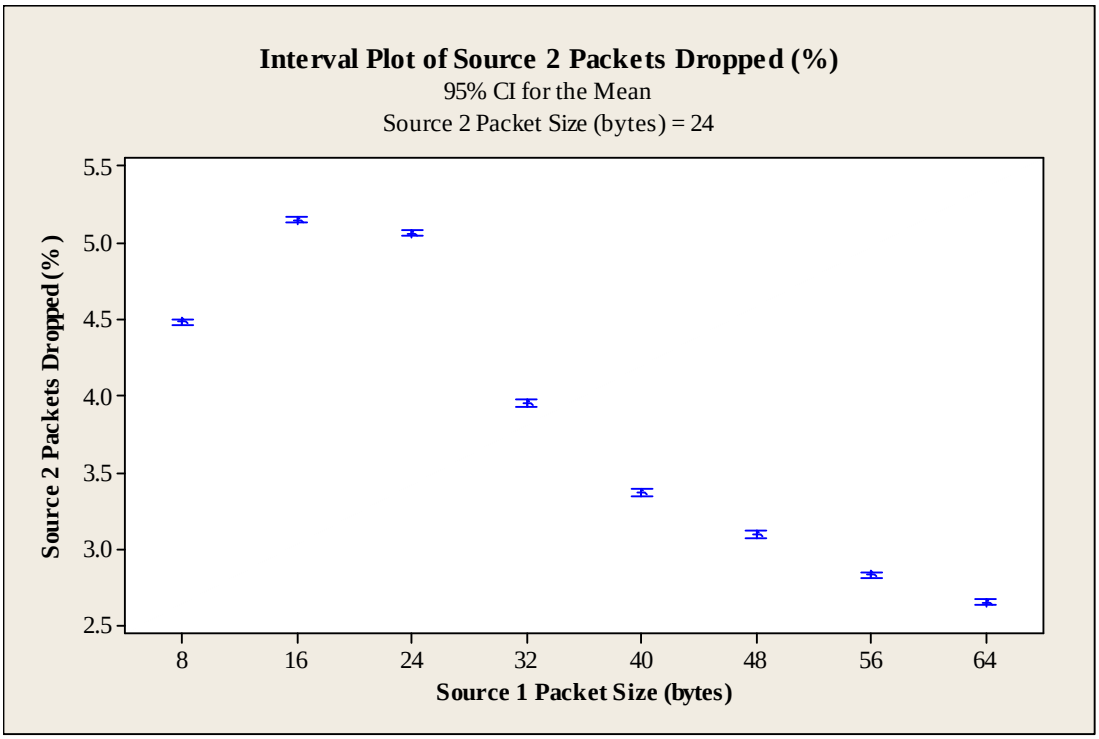

Figure 55: 95% CI for mean % of 24-B S2 packets dropped in Scenario 1 (no NTO)

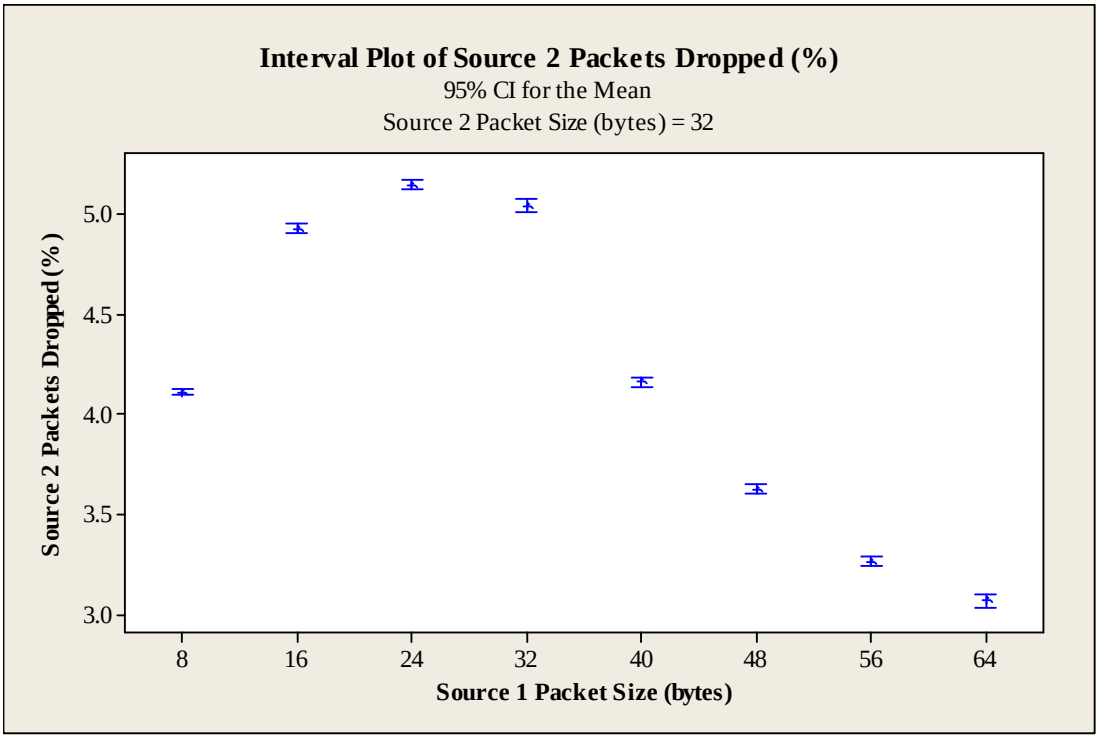

Figure 56: 95% CI for mean % of 32-B S2 packets dropped in Scenario 1 (no NTO)

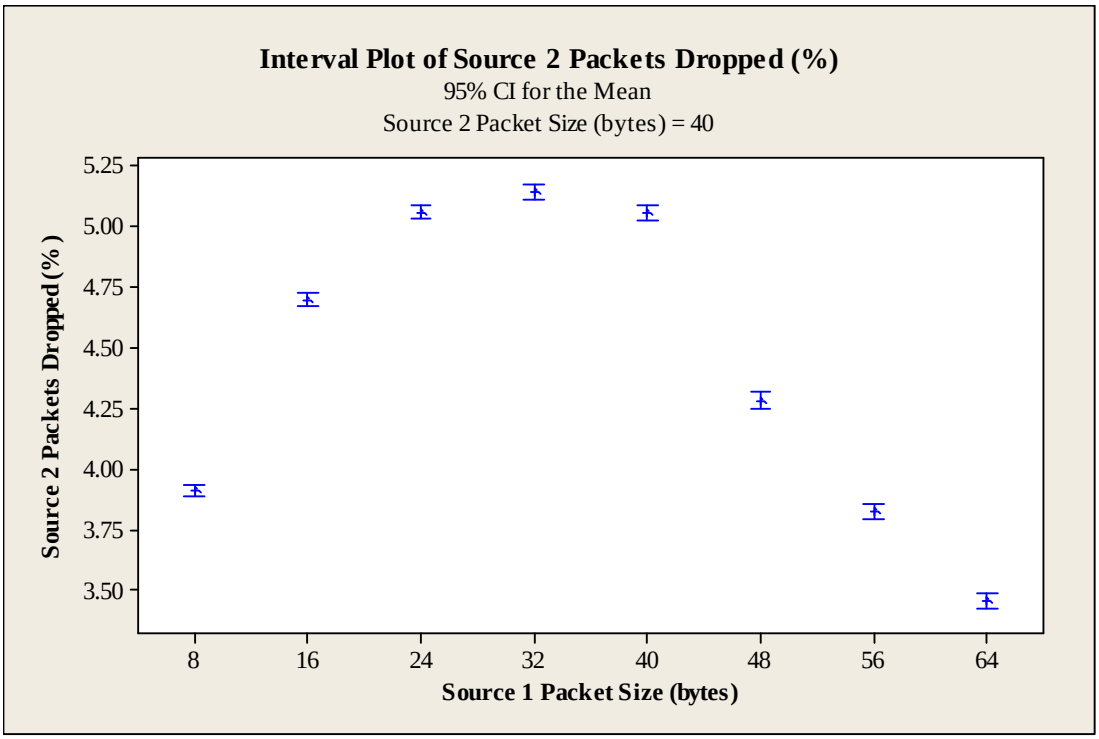

Figure 57: 95% CI for mean % of 40-B S2 packets dropped in Scenario 1 (no NTO)

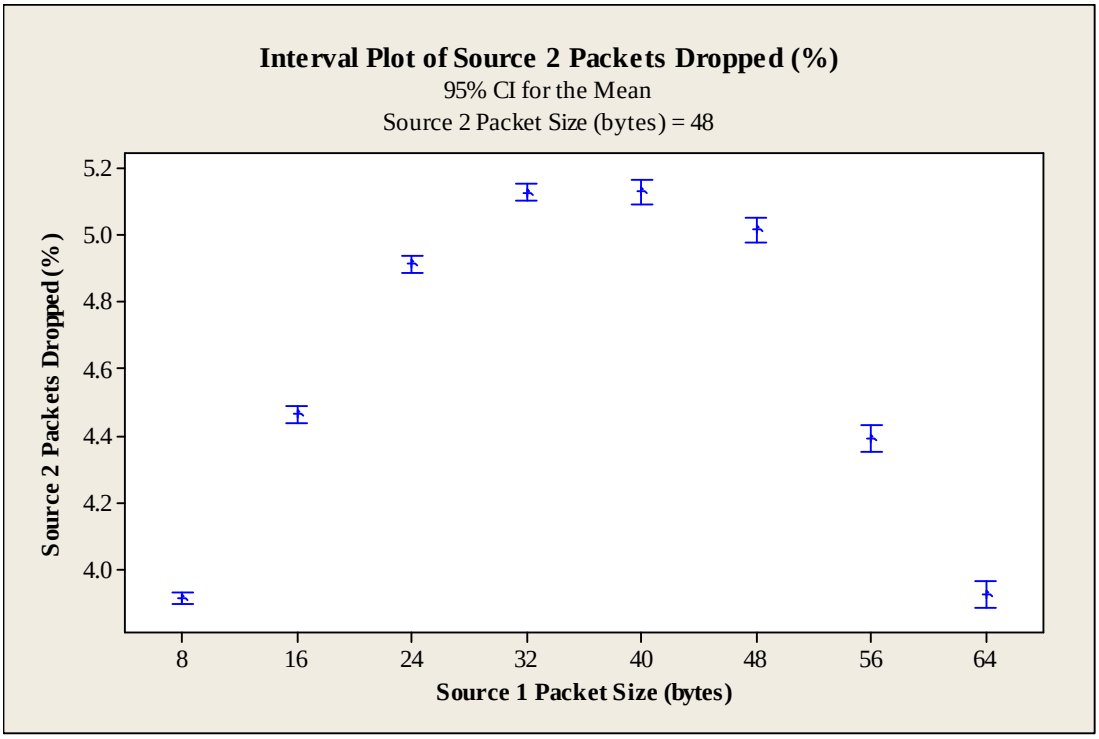

Figure 58: 95% CI for mean % of 48-B S2 packets dropped in Scenario 1 (no NTO)

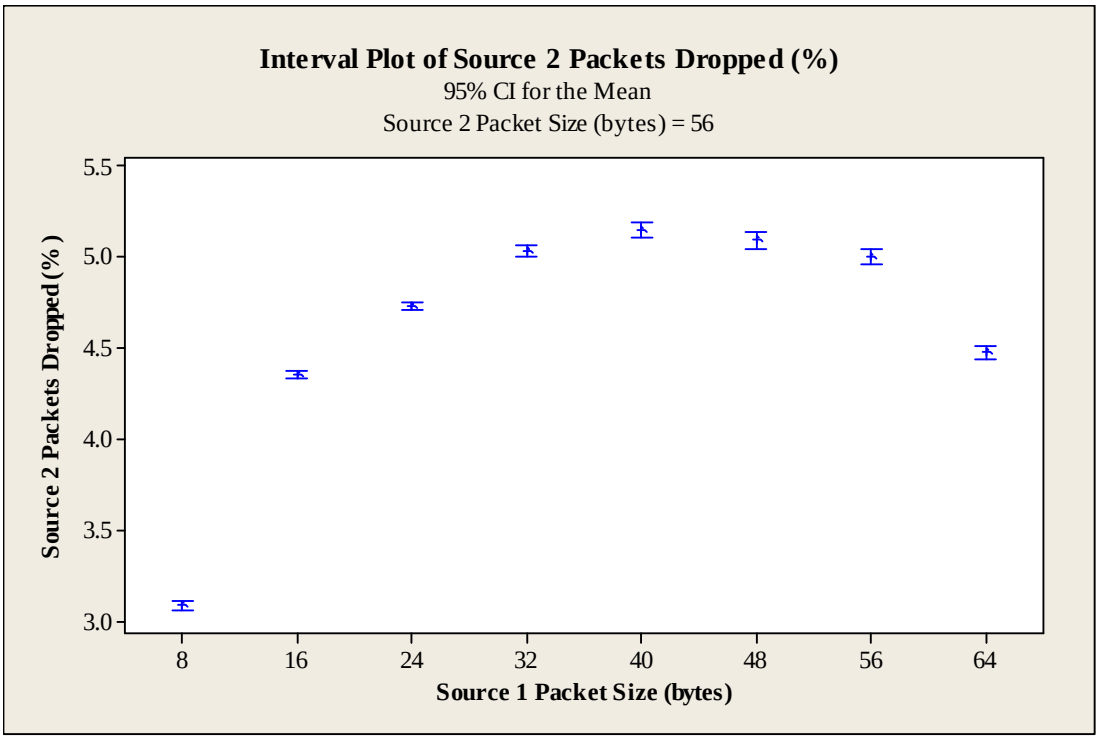

Figure 59: 95% CI for mean % of 56-B S2 packets dropped in Scenario 1 (no NTO)

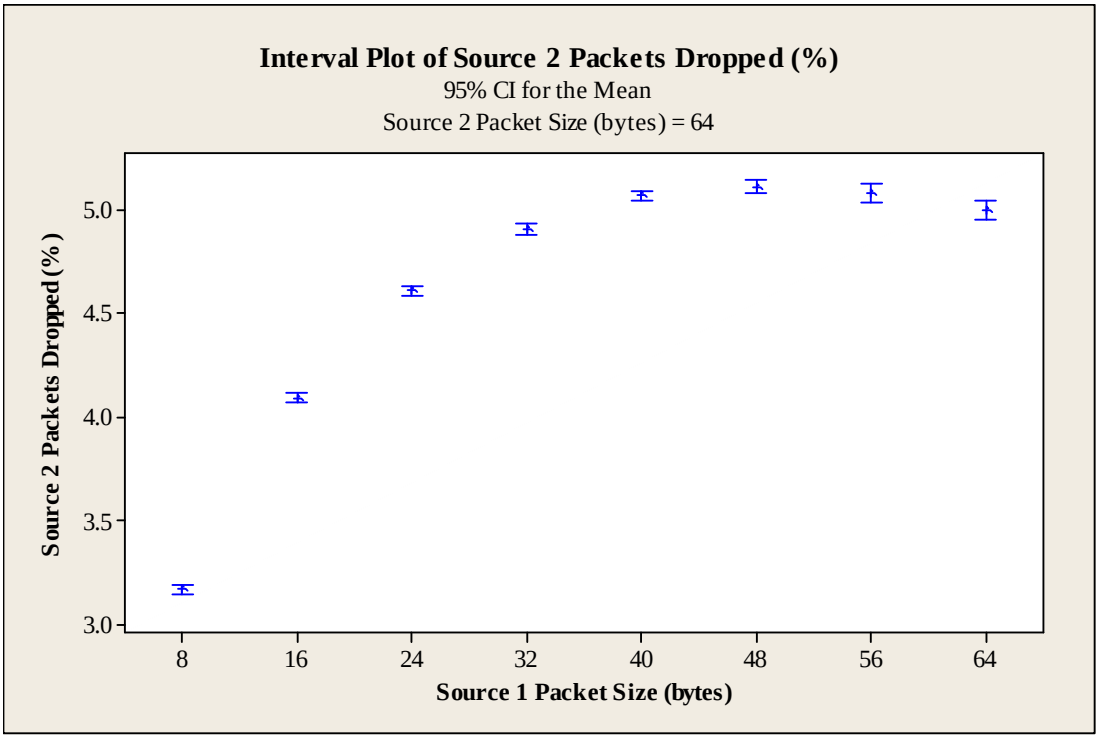

Figure 60: 95% CI for mean % of 64-B S2 packets dropped in Scenario 1 (no NTO)

Appendix J. Plots of Δ

The following plots illustrate the variation of Δ across ten polymorphisms for all test cases run. In each plot, the panels labeled 0 – 29 show the results from the thirty randomly generated input files. Each plot is titled with the name of the test case, where the first number indicates the number of nodes, the second number indicates the number of commodities, and the third number indicates the number of interface types. The values on the horizontal axes are for polymorphism number. The first solution is polymorphism 0 and has no measured difference, thus it is not plotted. The next nine polymorphisms each have the difference Δ measured from the previous polymorphism. The values on the vertical axes show the magnitude of this measured difference. Non-zero values for Δ designate that a polymorphism is indeed different, and larger values indicate a larger difference.

Since the set of possible edges is restricted to 25% incidence, there are a few cases where there are not many options for polymorphism, especially in the cases for five nodes. As a result, for these cases, polymorphisms oscillate among a few solutions. This behavior is reflected in some of these plots as periodicity. However, it must be noted that periodicity in these graphs does not imply oscillation between polymorphisms. It is possible to contrive scenarios where Δ fluctuates among a set of fixed values, but every polymorphism is distinct.

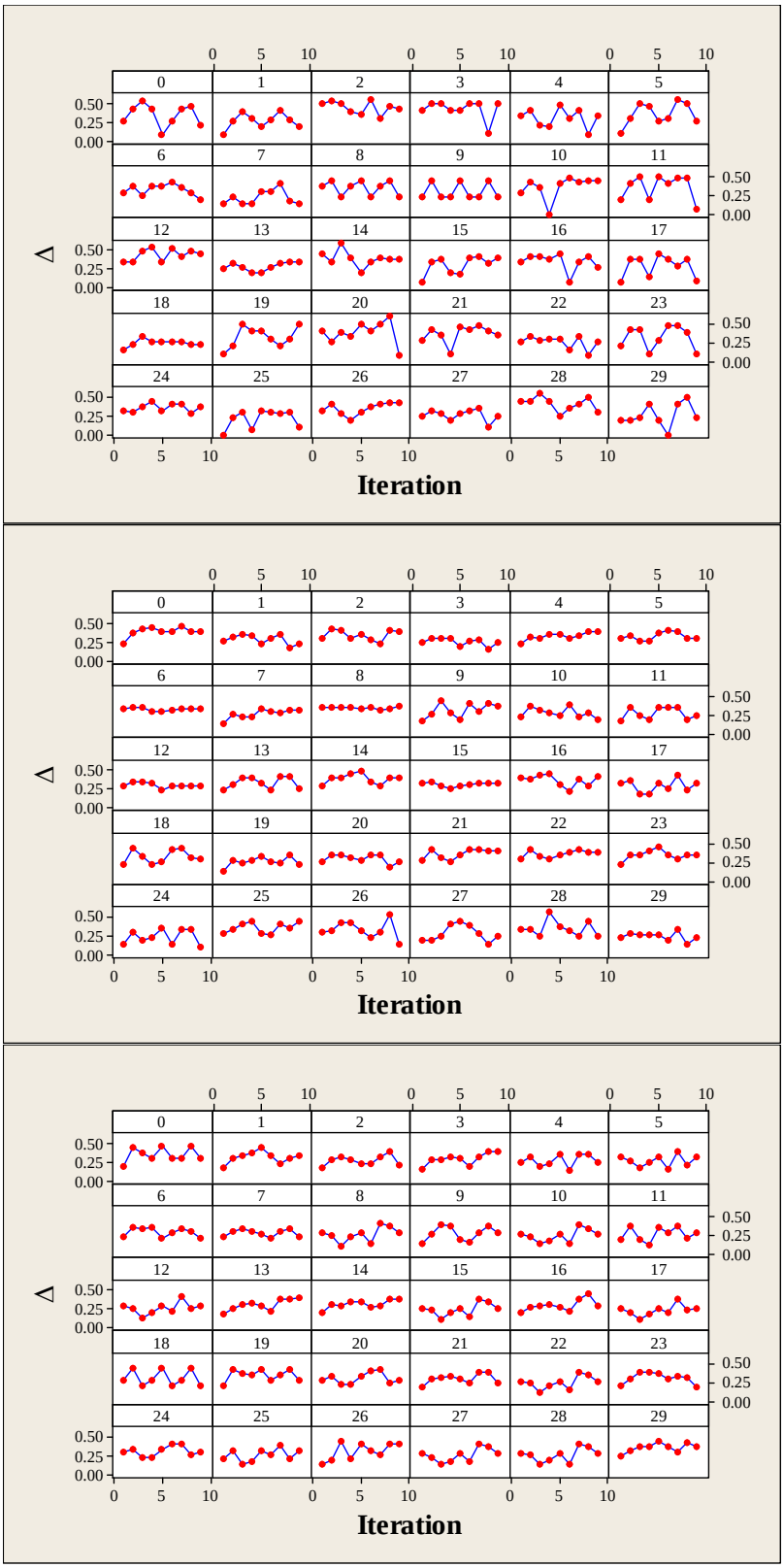

Figure 61: Plots of Δ by polymorphism for 5N1C1I, 5N2C1I, and 5N3C1I

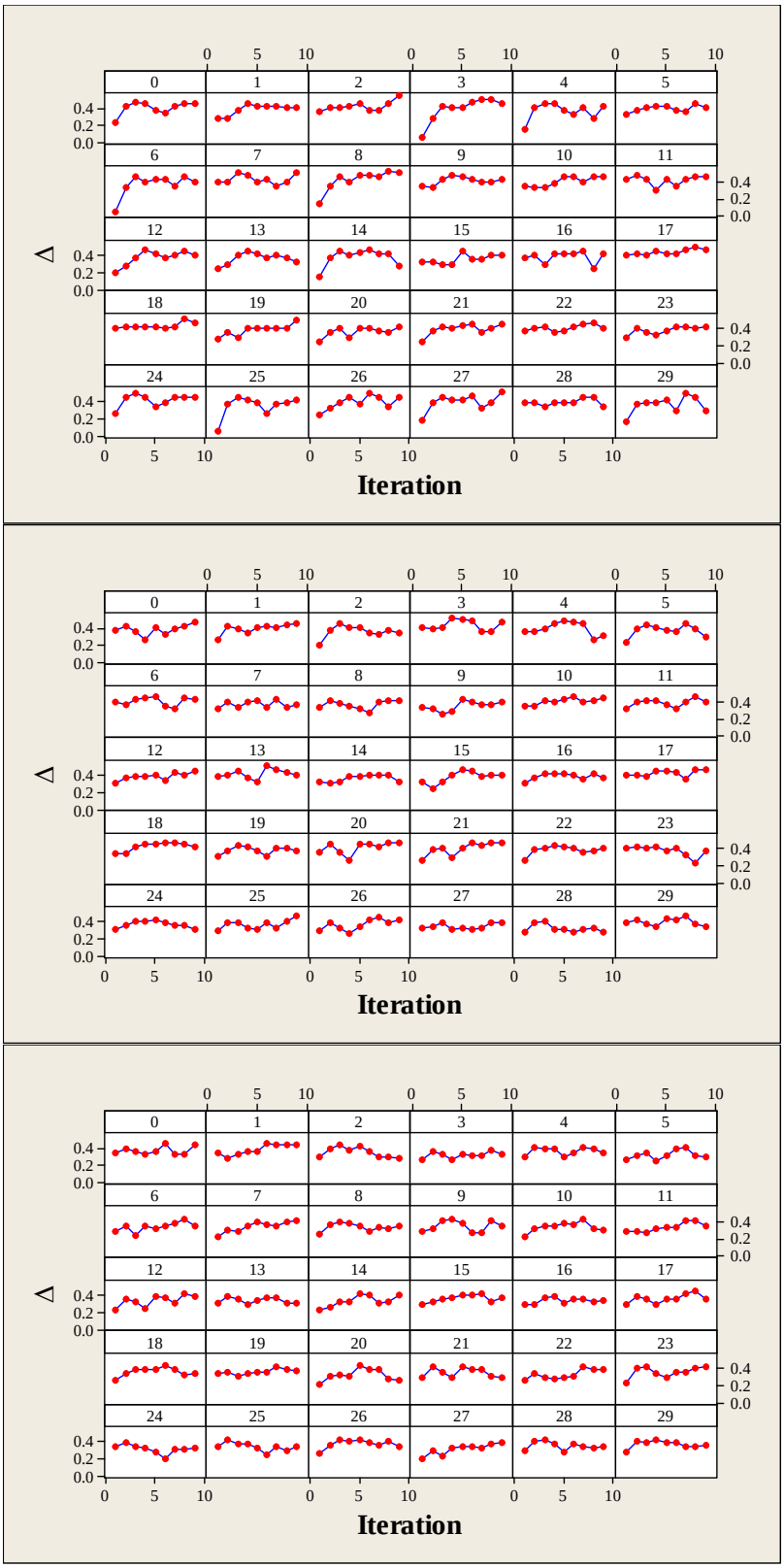

Figure 62: Plots of Δ by polymorphism for 5N1C2I, 5N2C2I, and 5N3C2I

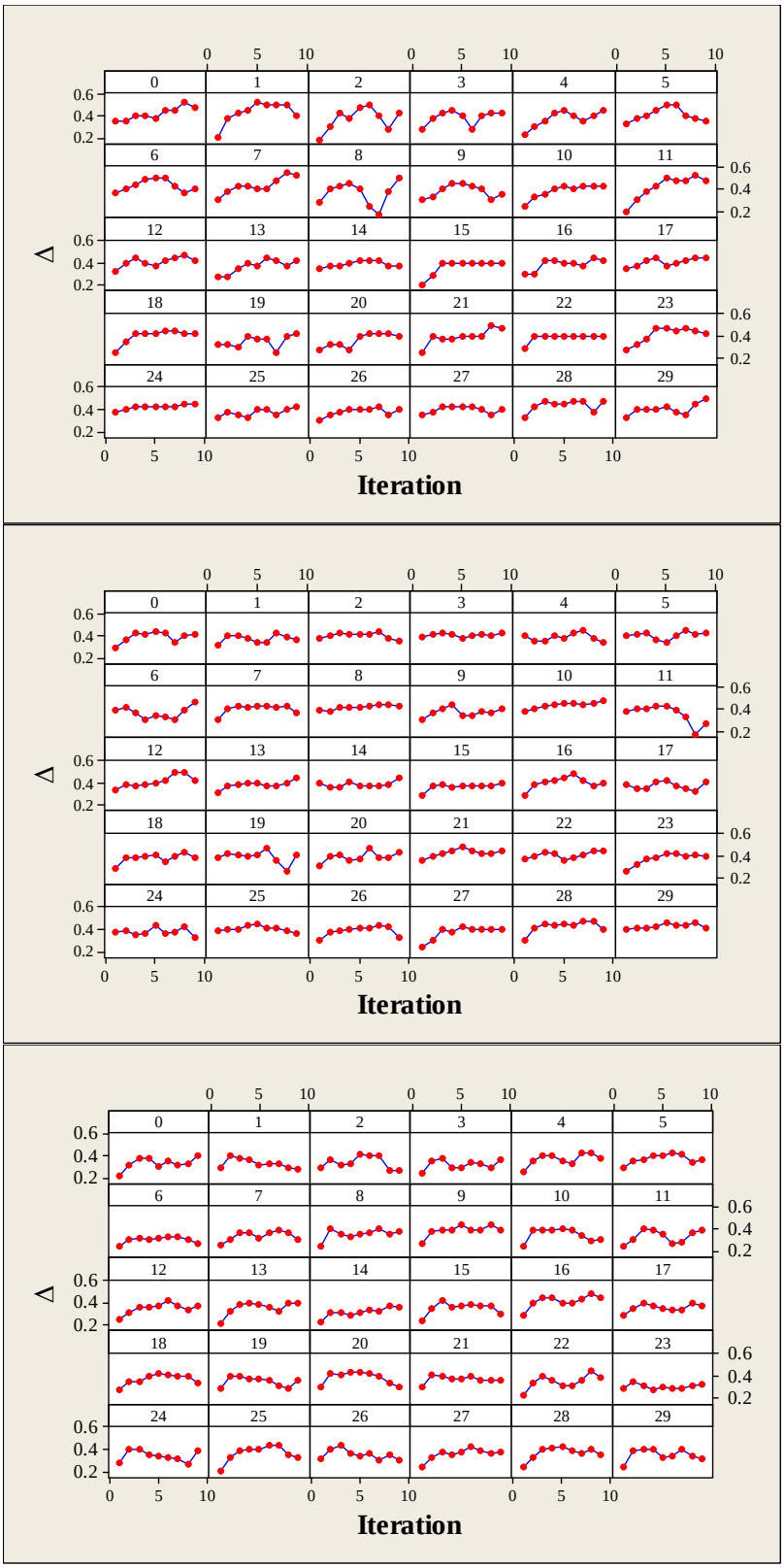

Figure 63: Plots of Δ by polymorphism for 5N1C3I, 5N2C3I, and 5N3C3I

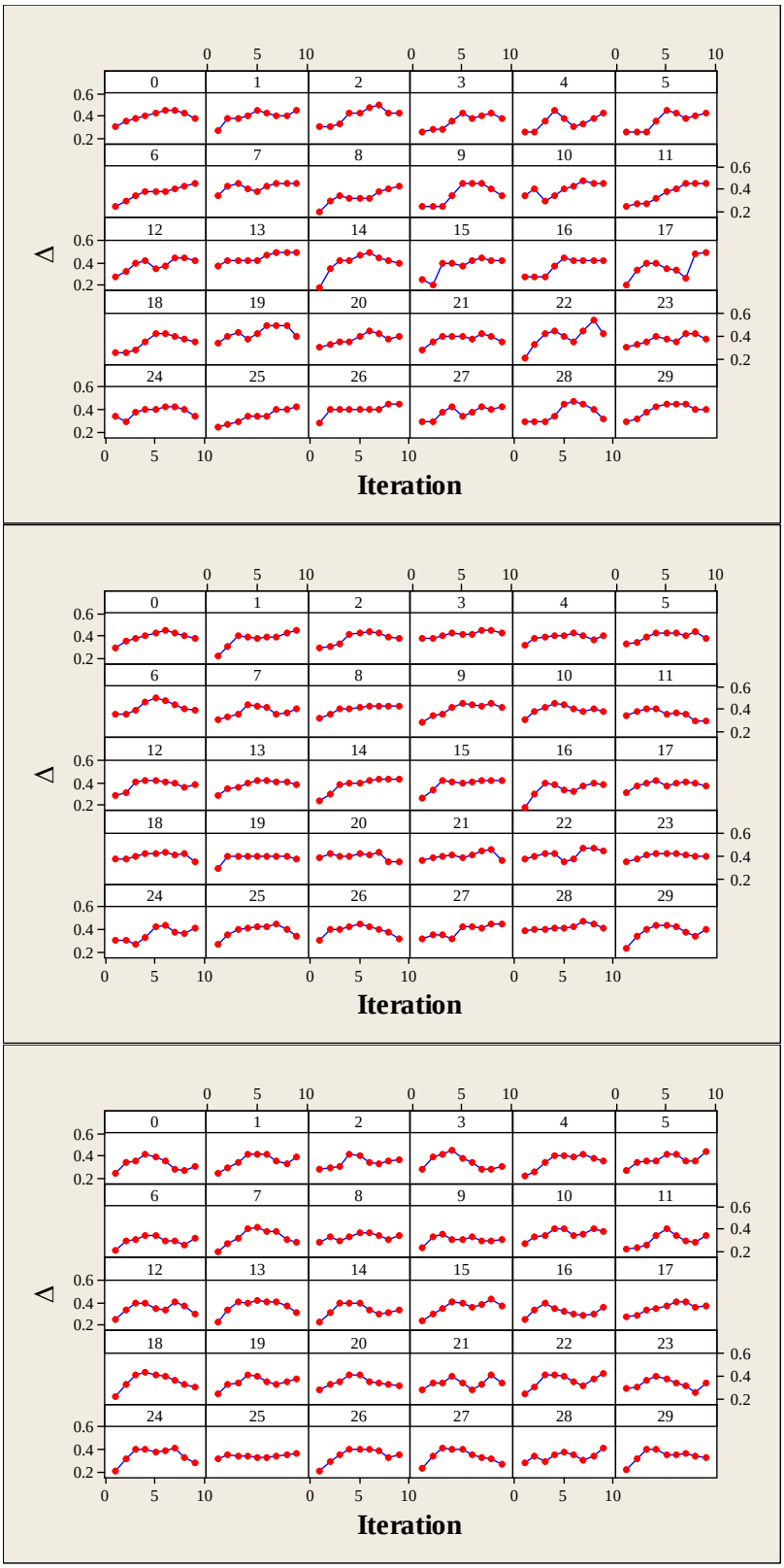

Figure 64: Plots of Δ by polymorphism for 5N1C4I, 5N2C4I, and 5N3C4I

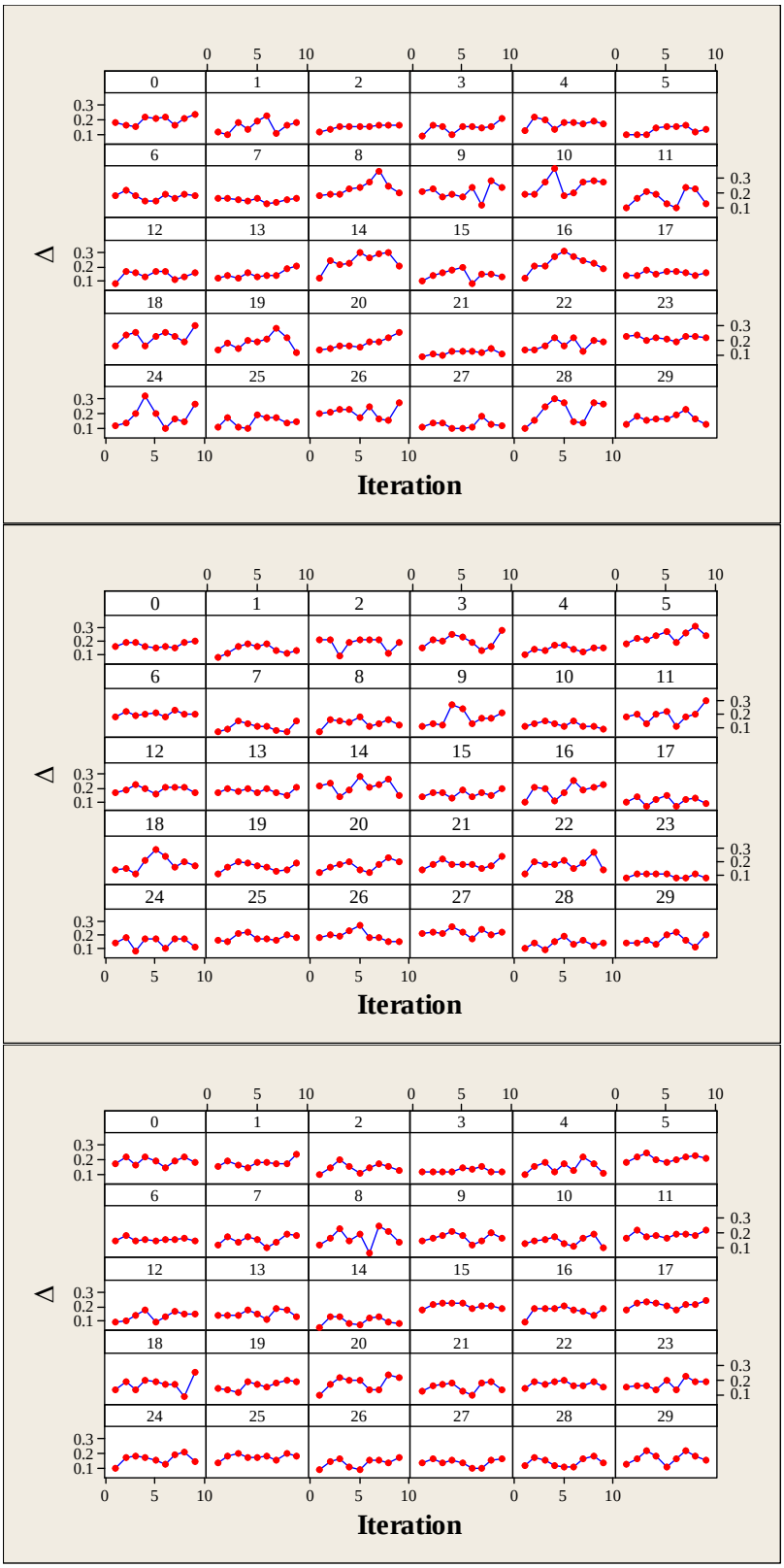

Figure 65: Plots of Δ by polymorphism for 10N1C1I, 10N2C1I, and 10N3C1I

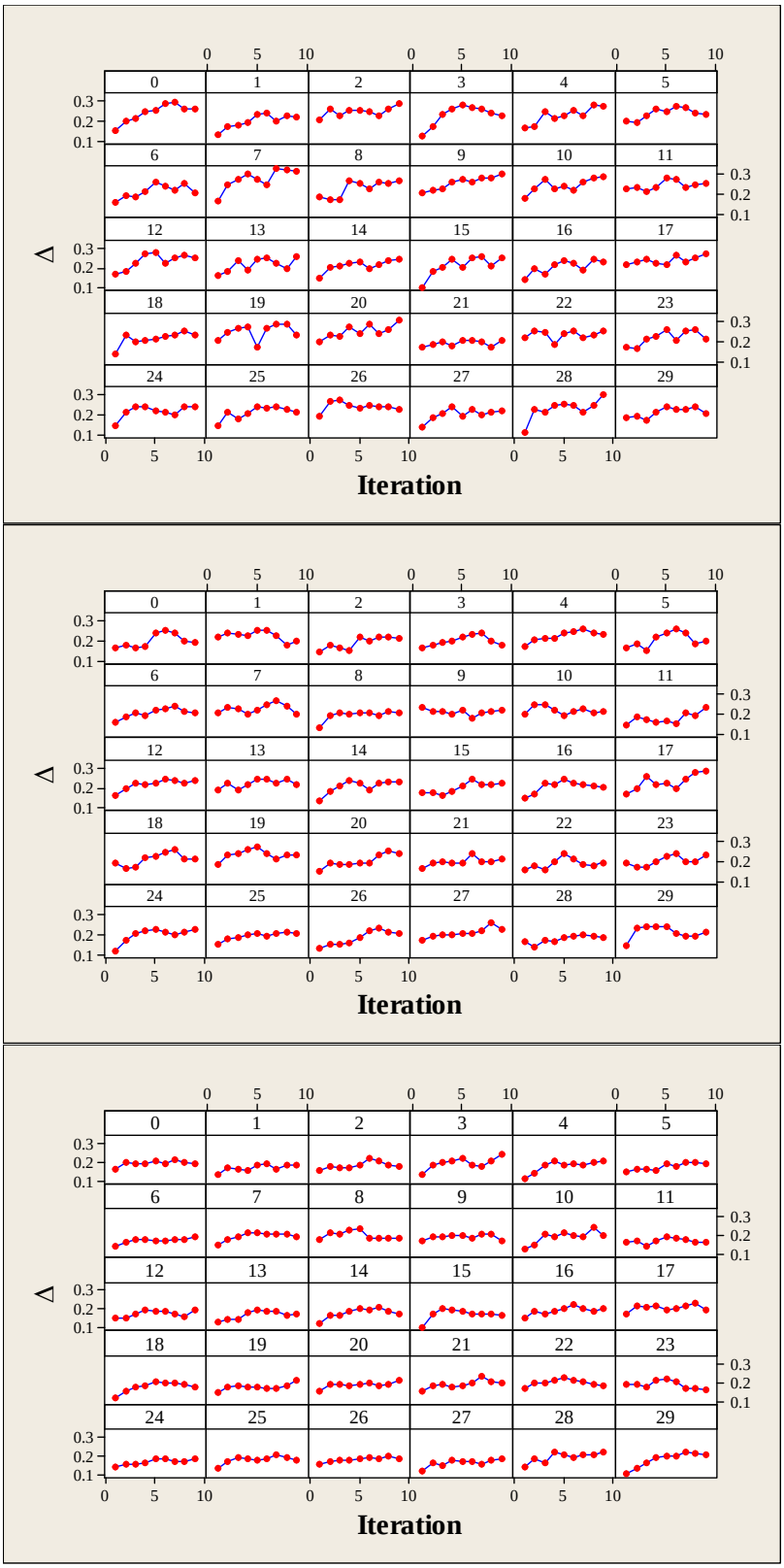

Figure 66: Plots of Δ by polymorphism for 10N1C2I, 10N2C2I, and 10N3C2I

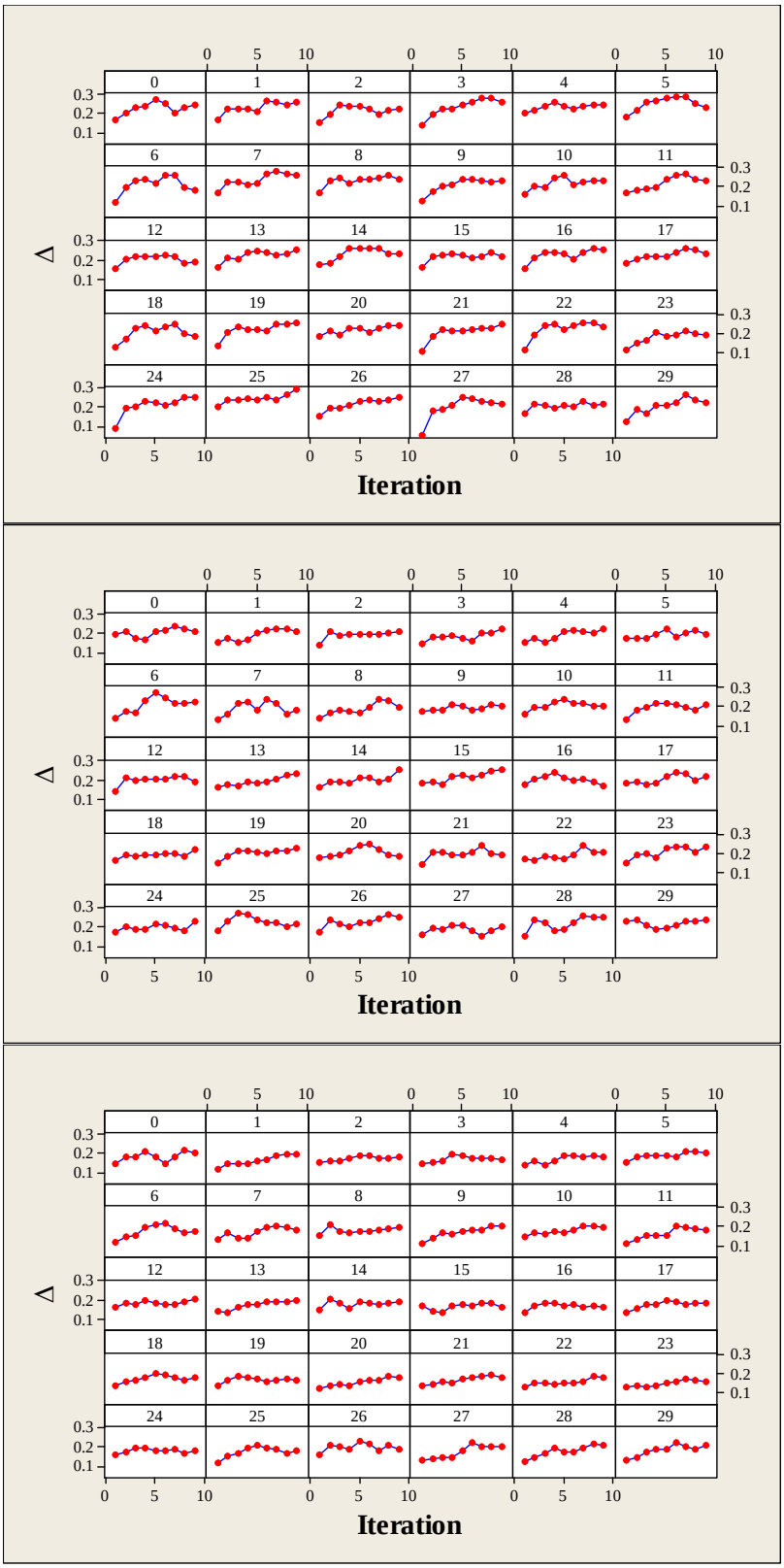

Figure 67: Plots of Δ by polymorphism for 10N1C3I, 10N2C3I, and 10N3C3I

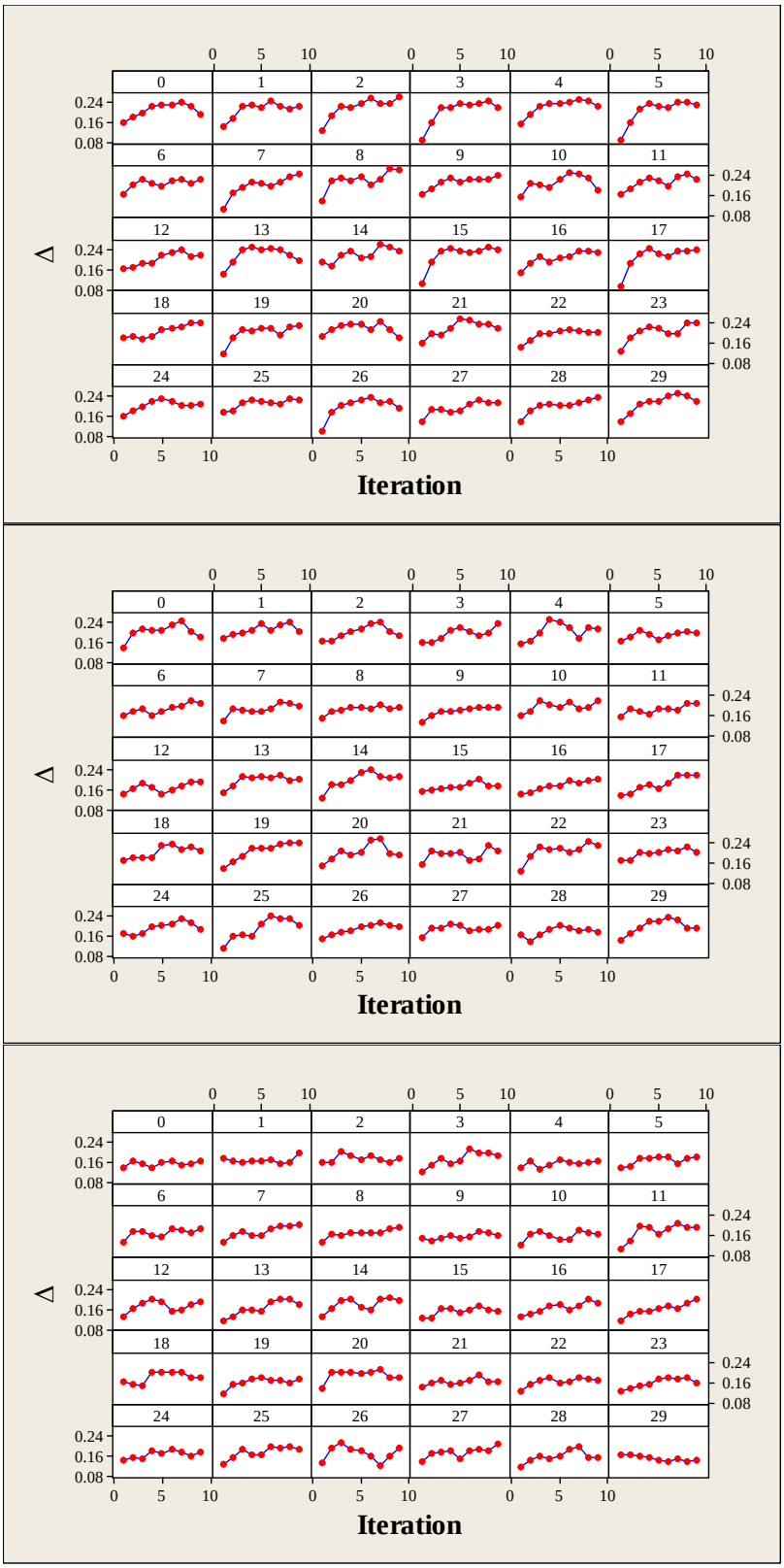

Figure 68: Plots of Δ by polymorphism for 10N1C4I, 10N2C4I, and 10N3C4I

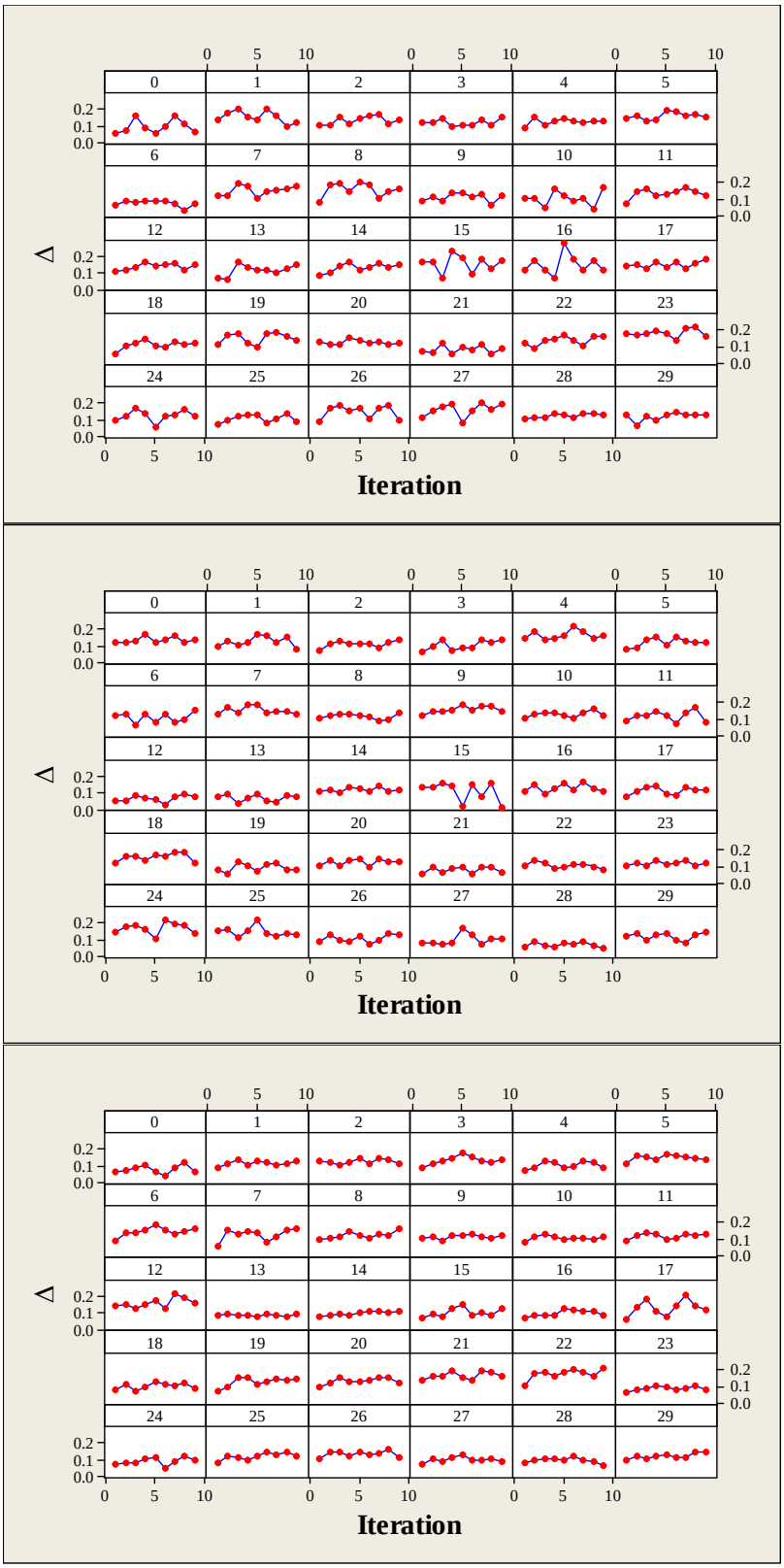

Figure 69: Plots of Δ by polymorphism for 15N1C1I, 15N2C1I, and 15N3C1I

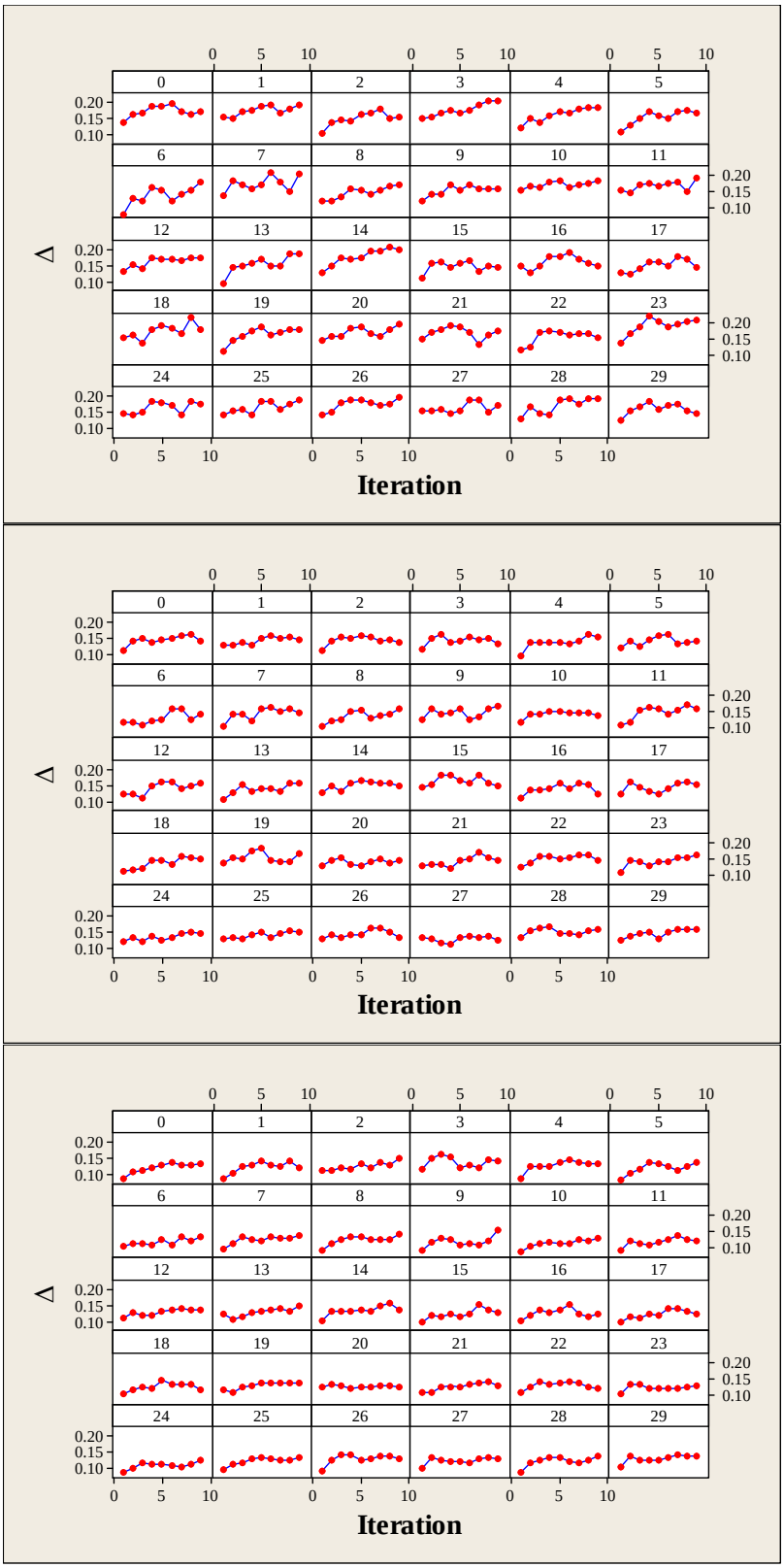

Figure 70: Plots of Δ by polymorphism for 15N1C2I, 15N2C2I, and 15N3C2I

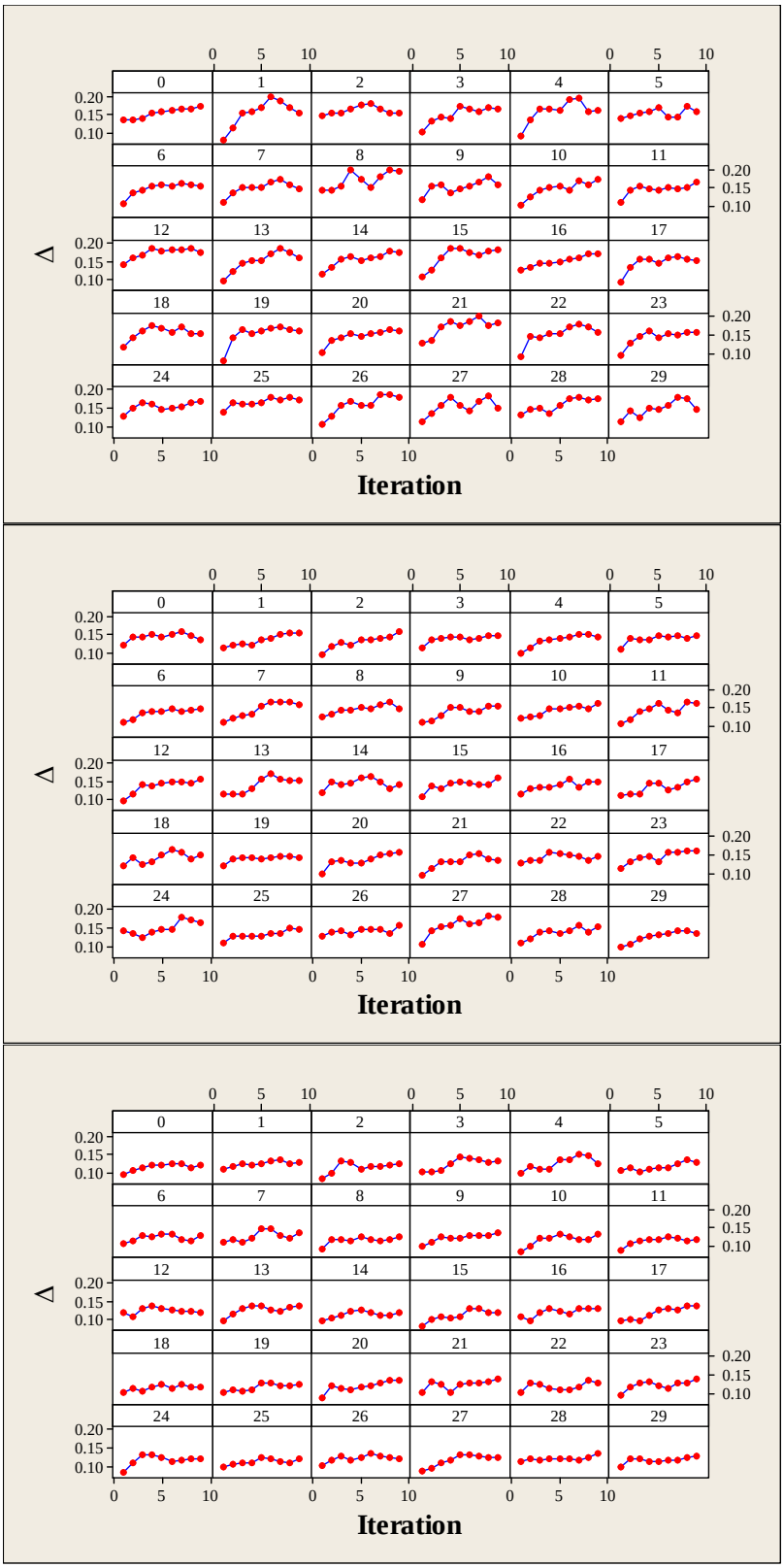

Figure 71: Plots of Δ by polymorphism for 15N1C3I, 15N2C3I, and 15N3C3I

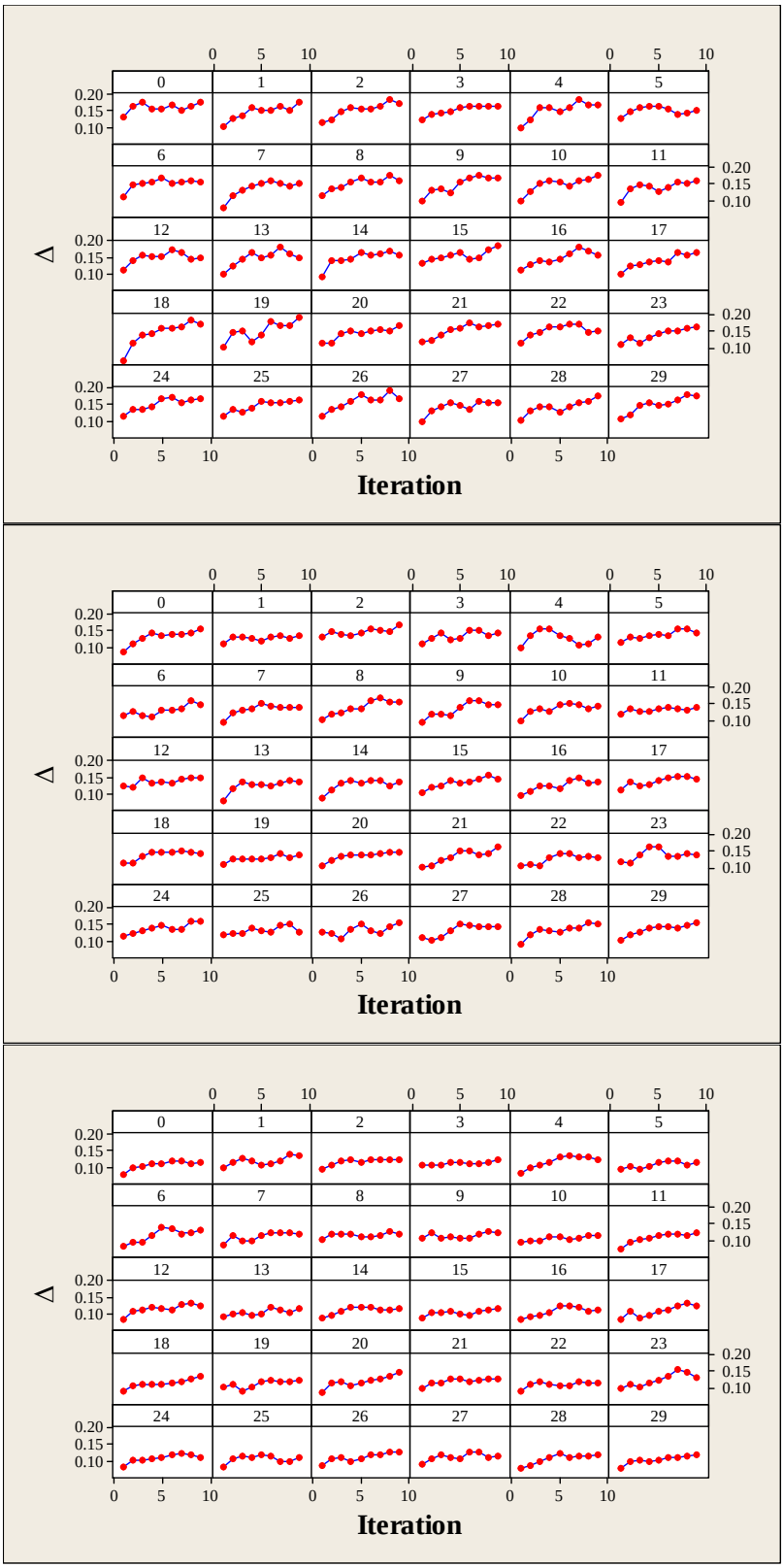

Figure 72: Plots of Δ by polymorphism for 15N1C4I, 15N2C4I, and 15N3C4I

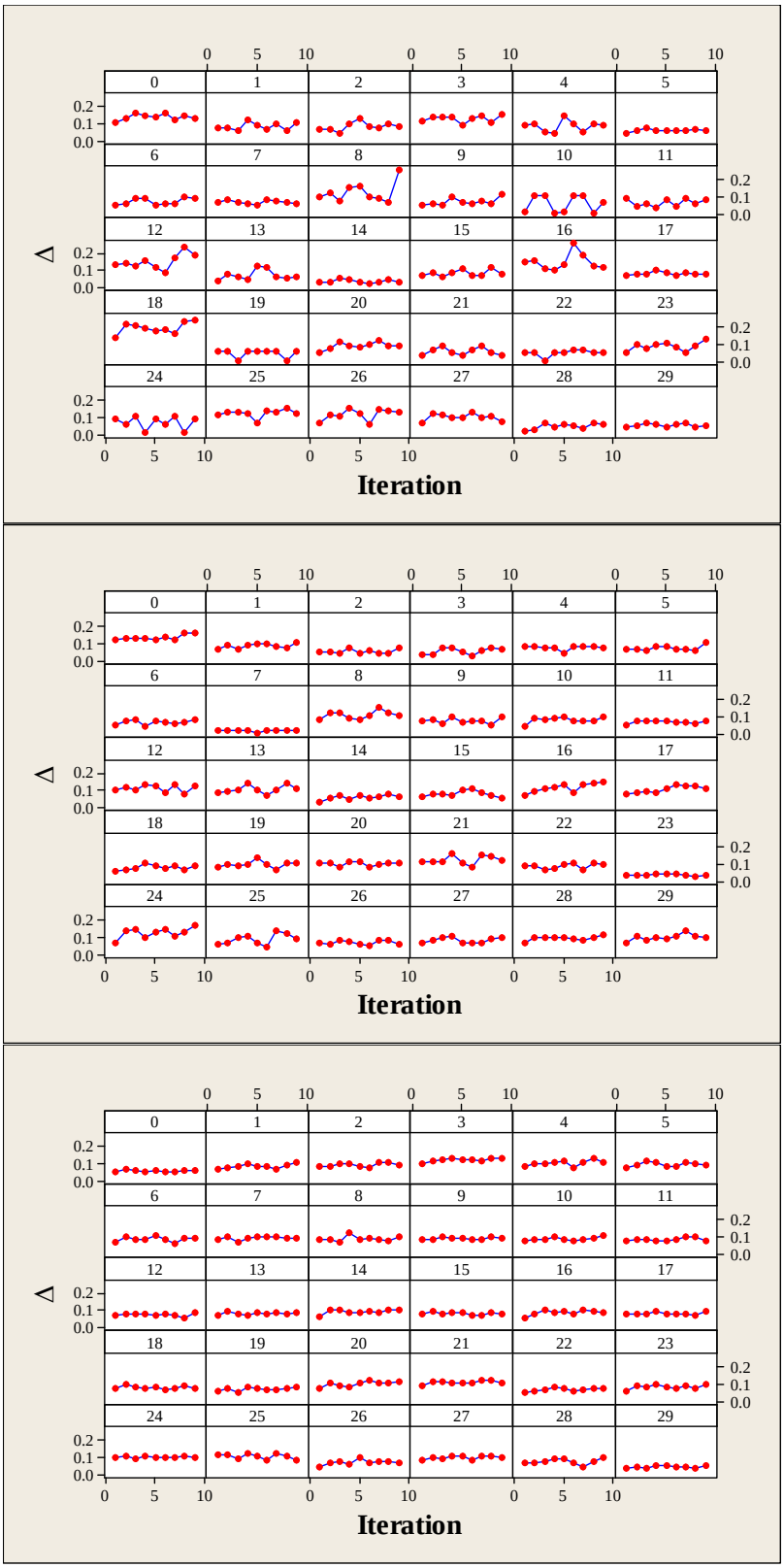

Figure 73: Plots of Δ by polymorphism for 20N1C1I, 20N2C1I, and 20N3C1I

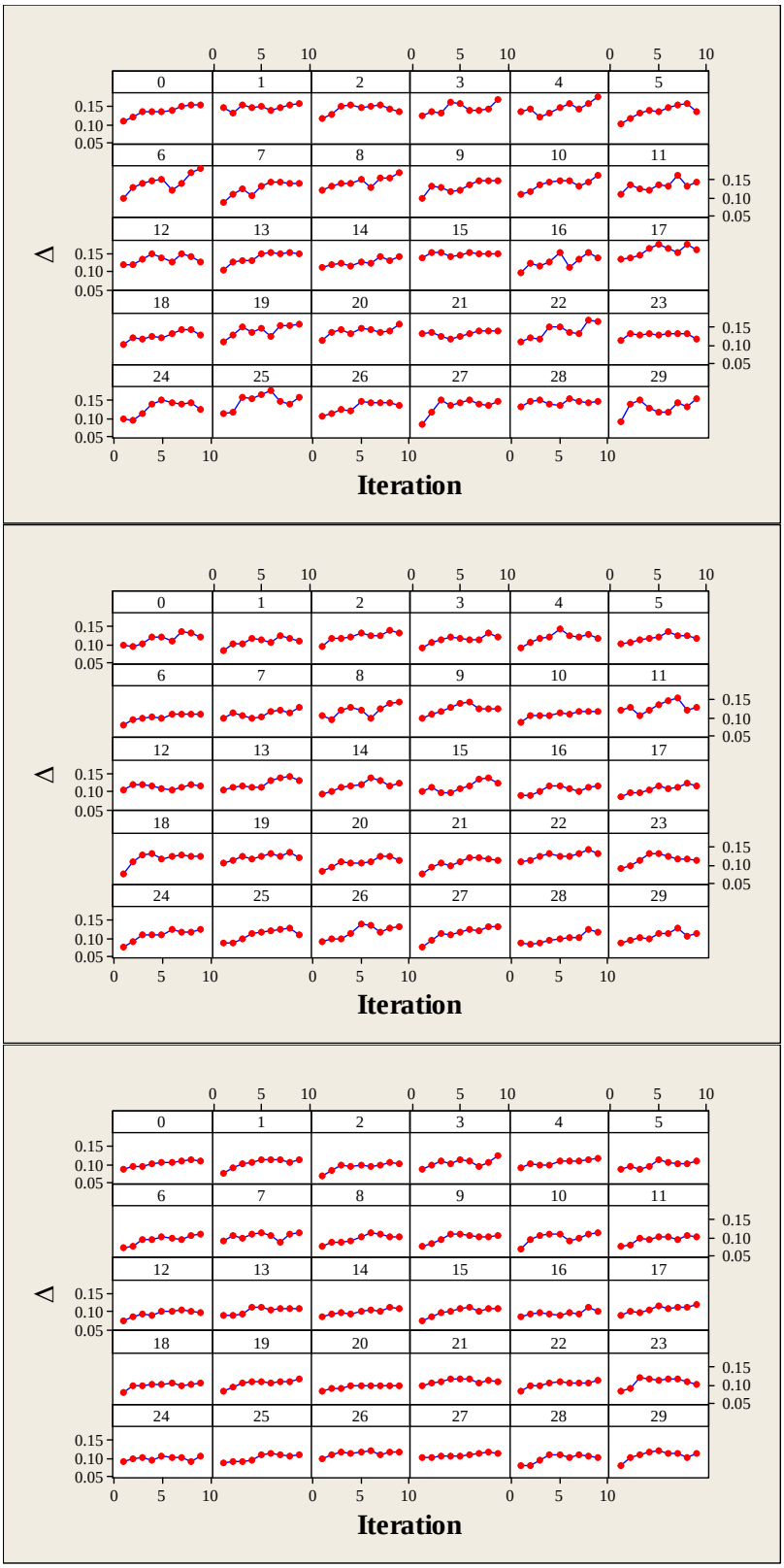

Figure 74: Plots of Δ by polymorphism for 20N1C2I, 20N2C2I, and 20N3C2I

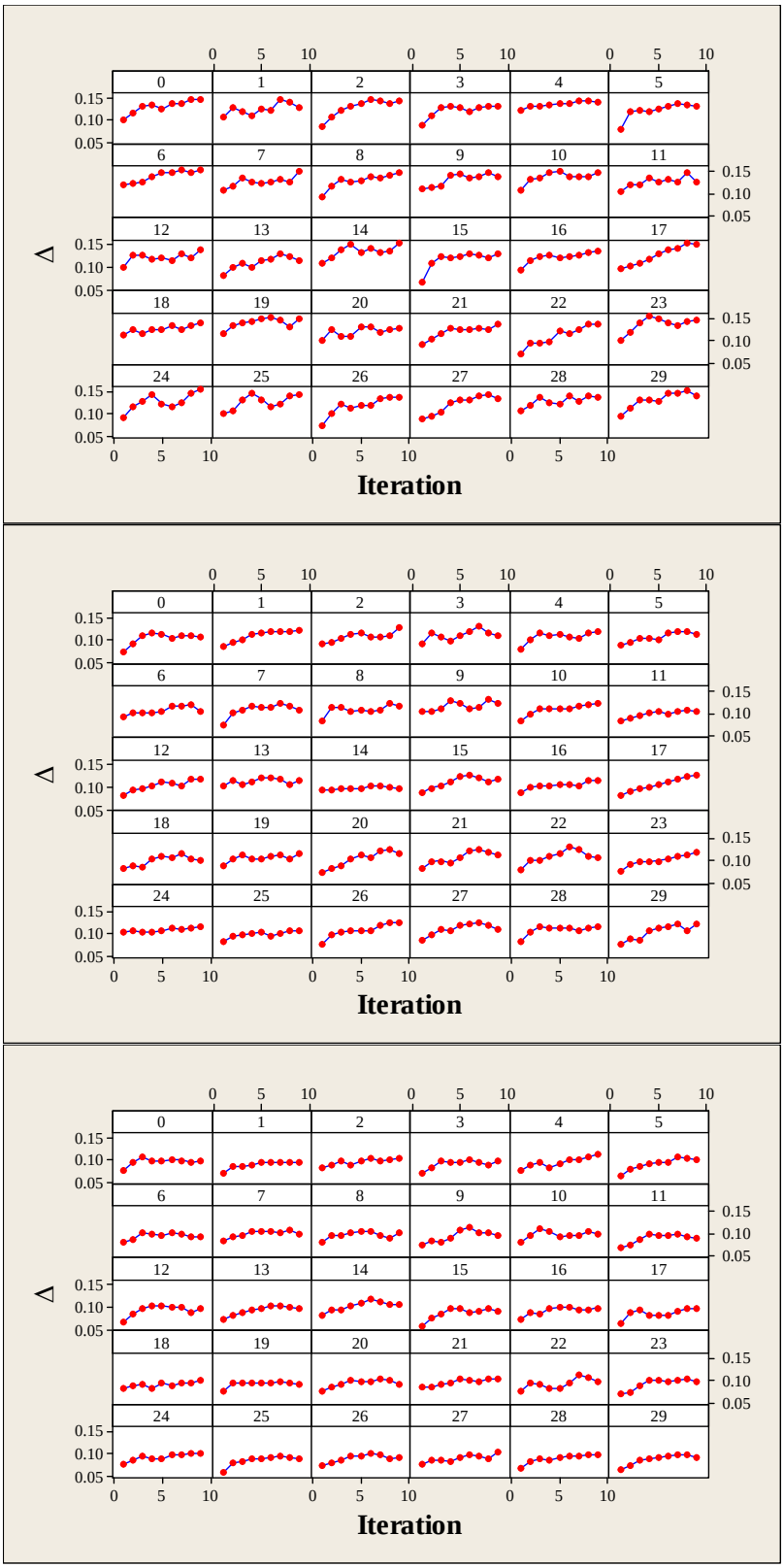

Figure 75: Plots of Δ by polymorphism for 20N1C3I, 20N2C3I, and 20N3C3I

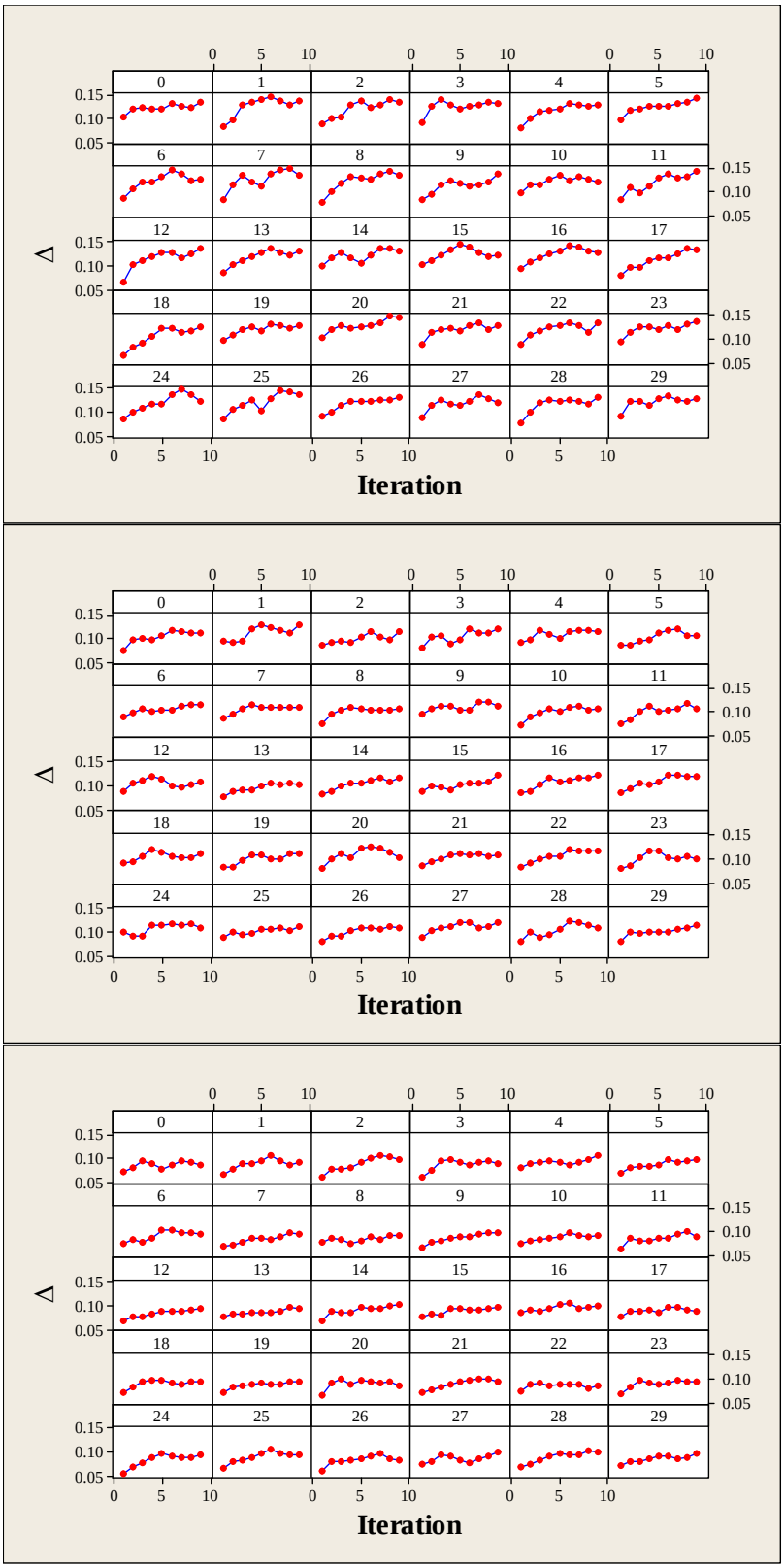

Figure 76: Plots of Δ by polymorphism for 20N1C4I, 20N2C4I, and 20N3C4I

Bibliography

[1] Department of Defense. *Global Information Grid (GIG) Overarching Policy*. DoD Directive 8100.1. 19 September 2002.

[2] Giordano, Silvia. "Mobile Ad Hoc Networks," in *Handbook of Wireless Networks and Mobile Computing*. Ed. Ivan Stojmenović. New York: John Wiley & Sons, Inc., 2002.

[3] Karimi, Masoumeh and Deng Pan, "Challenges for Quality of Service (QoS) in Mobile Ad-Hoc Networks (MANETs)," *Wireless and Microwave Technology Conference (WAMICON '09), Clearwater, FL, 20-21 April 2009.* 1-5. Piscataway NJ: IEEE Operations Center, 2009.

[4] Rajan, M. A., M. Girish Chandra, Lokanatha C. Reddy, and Prakash Hiremath. "A Study on Network partition Detection Relevant to Ad-hoc Networks: Connectivity Index Approach," *International Journal of Computer Science and Network Security (IJCSNS)*, 8: 150-158 (June 2008).

[5] DoD CIO. *Department of Defense Global Information Grid Architectural Vision: Vision for a Net-Centric, Service-Oriented DoD Enterprise*. June 2007.

[6] Camp, Tracy, Jeff Boleng, and Vanessa Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking – Research, Trends, and Applications*, 2: 483-502 (August 2002).

[7] Bettstetter, Christian, Giovanni Resta, and Paolo Santi. "The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, 2: 257-269 (July-September 2003).

[8] Kleeman, Mark P., Gary B. Lamont, Kenneth M. Hopkinson, and Scott R. Graham. "Solving Multicommodity Capacitated Network Design Problems using a Multiobjective Evolutionary Algorithm," *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2007), Honolulu, 1-5 April 2007*. 33-41. New York: IEEE, 2007.

[9] Garner, Roger L. *Heuristically Driven Search Methods for Topology Control in Directional Wireless Hybrid Networks*. MS thesis, AFIT/GCS/ENG/07-03. Graduate School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 2007 (ADA469317).

[10] Oimoen, Steven C. *Dynamic Network Formation Using Ant Colony Optimization*. PhD dissertation, AFIT/DCS/ENG/09-06. Graduate School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 2009 (ADA495713).

[11] Ranne, Wayne K. and Larry K. McKee, Jr. "Global Information Grid NetOps Tasking Orders (GNTO) White Paper." Smithfield VA: Select Innovation, 28 June 2006. Web. 4 March 2010.

[12] McKee, Larry K., Jr. "RE: AFIT researchers interested in your whitepaper." Message to Vinod Naga. n. pag. 9 July 2009. E-mail.

[13] Stookey, David E. *A Notional Battlespace for Simulating and Testing Dynamic Wireless Networks*. Graduate Research Project, AFIT/ENG/IC4-06J. Graduate School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, June 2006 (ADA453676).

[14] Pecarina, John M. *Creating an Agent Based Framework to Maximize Information Utility*. MS thesis, AFIT/GCS/EMG/08-19. Graduate School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 2008 (ADA482972).

[15] Heinke, Ward E., AFNOC Director. "AFNETOPS Direction." Address to ACC Commanders' Conference, Offutt AFB NE, 6-8 May 2008. PowerPoint slides.

[16] Pistilli, David B. *United States Air Force Network Operations Functional Concept*. HQ 8 AF/Det 1, 12 October 2006.

[17] Imperial, Matthew J. "RE: Question about the 24th AF." Message to Matthew Compton. n. pag. 20 January 2010. E-mail.

[18] Weida, Johnny A. *Air Force Guidance Memorandum on Command and Control (C2) of the AF-GIG*. AFGM13-01. n. pag. HQ USAF/A3/5, Washington, 1 September 2009.

[19] Imperial, Matthew J. "RE: Question about the 24th AF." Message to Matthew Compton. n. pag. 18 December 2009. E-mail.

[20] Scott, William B. and Craig Covault. "High Ground Over Iraq," *Aviation Week*. The McGraw-Hill Companies, 8 June 2003. n. pag. Web. 8 March 2010.

[21] Joint Chiefs of Staff. *Doctrine for Joint Operations.* Joint Publication (JP) 3-0. Washington: GPO, 10 September 2001.

[22] 705 Training Squadron. *Air and Space Operations Center*. Air Force Operational Tactics, Techniques, and Procedures (AFOTTP) 2-3.2. Washington: GPO, 13 December 2004.

[23] Joint Chiefs of Staff. *Department of Defense Dictionary of Military and Associated Terms*, Joint Publication (JP) 1-02. Washington: GPO, 12 April 2001 (As Amended Through 13 June 2007).

[24] Deputy Chief of Staff, Plans and Operations, Headquarters USAF. *JFACC Primer*, Washington: HQ USAF/XOXD, January 10, 1994.

[25] Air Force Doctrine Center. *Leadership and Force Development*, Air Force Doctrine Document (AFDD) 1-1. Maxwell-Gunter AFB AL: AFDC, 18 February 2006.

[26] Joint Chiefs of Staff. *Command and Control for Joint Air Operations*, Joint Publication (JP) 3-56.1. Washington: GPO, 14 November 1994.

[27] Joint Doctrine & Concepts Centre. *Joint Air Operations*, Interim Joint Warfare Publication (IJWP) 3-30. Shrivenham UK: JDCC, October 2003.

[28] The MITRE Corporation. *USMTF Message Brower Help*. Version: USMTF 2004 Baseline. Computer software. Falls Church VA: DISA/Center for Joint & Coalition Interoperability, 2004. Microsoft Windows Help File.

[29] Santi, Paolo. *Topology Control in Wireless Ad Hoc and Sensor Networks*. West Sussex: John Wiley & Sons Ltd., 2005.

[30] Rajaraman, Rajmohan. "Topology Control and Routing in Ad Hoc Networks: A Survey," *SIGACT News*, 33: 60-73 (June 2002).

[31] Ahuja, Ravindra K., Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ: Prentice-Hall, 1993.

[32] Kumar, Umesh, Himanshu Gupta, and Samir R. Das. "A Topology Control Approach to Using Directional Antennas in Wireless Mesh Networks," *Proceedings IEEE International Conference on Communications (ICC), Istanbul, 11-15 June 2006*. 4083-4088. Piscataway NJ: IEEE Operations Center, 2006.

[33] Namboodiri, Vinod, Lixin Gao, and Ramakrishna Janaswamy. “Power-efficient topology control for static wireless networks with switched beam directional antennas,” *Ad Hoc Networks,* 6: 287-306 (April 2008).

[34] Erwin, Michael C. *Combining Quality of Service and Topology Control in Directional Hybrid Wireless Networks*. MS thesis, AFIT/GOR/ENS/06-07. Graduate School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 2006 (ADA445194).

[35] Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms* (2nd Edition). Cambridge, MA: The MIT Press, 2001.

[36] Kleinberg, Jon and Éva Tardos. *Algorithm Design*. Boston: Pearson Education, 2006.

[37] Davis, Christopher C., Igor I. Smolyaninov, and Stuart D. Milner. “Flexible Optical Wireless Links and Networks,” *IEEE Communications Magazine*, 41: 51-57 (March 2003).

[38] Desai, Aniket and Stuart D. Milner. “Autonomous Reconfiguration in Free-Space Optical Sensor Networks,” *IEEE Journal on Selected Areas in Communication*, 23: 1556-1563 (August 2005).

[39] Lala, Jaynarayan, Douglas Maughan, Catherine McCollum, and Brian Witten. “Foreword,” *DARPA Information Survivability Conference & Exposition II: proceedings (DISCEX’01), Anaheim, 12-14 June 2001*. 0:viii-xi. Los Alamitos CA: IEEE Computer Society, 2001.

[40] Lowry, John and Kenneth Theriault. “Experimentation in the IA Program,” *DARPA Information Survivability Conference & Exposition II: proceedings (DISCEX’01), Anaheim, 12-14 June 2001*. 1:134-140. Los Alamitos CA: IEEE Computer Society, 2001.

[41] Kewley, Dorene, Russ Fink, John Lowry, and Mike Dean. “Dynamic Approaches to Thwart Adversary Intelligence Gathering,” *DARPA Information Survivability Conference & Exposition II: proceedings (DISCEX’01), Anaheim, 12-14 June 2001*. 1:176-185. Los Alamitos CA: IEEE Computer Society, 2001.

[42] McClure, Stuart, Joel Scambray, and George Kurtz. *Hacking Exposed: Network Security Secrets & Solutions* (5th Edition). Emeryville CA: McGraw Hill/Osborne, 2005.

[43] Krishnamurthy, Balachander, Harsha V. Madhyastha, and Suresh Venkatasubramanian. "On stationarity in Internet measurements through an information-theoretic lens," *Proceedings of the 21st International Conference on Data Engineering (ICDE '05), Tokyo, 5-8 April 2005*. 1185-1185. IEEE, 2005.

[44] Li, Lun, David Alderson, Walter Willinger, and John Doyle. "A First-Principles Approach to Understanding the Internet's Router-level Topology," *Proceedings of the ACM SIGCOMM 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'04), Portland, 30 August-3 September 2004*. 3-14. New York: Association for Computing Machinery, 2004.

[45] Harrington, Edward F. "Measuring network change: Rényi cross entropy and the second order degree distribution," *Proceedings of the Passive and Active Measurement Conference (PAM 2006), Adelaide, Australia, 30-31 March 2006*. 161-170. Web. 12 March 2010.

[46] Defense Information Systems Agency. *JSC – Information Systems*. n. pag. Web. 15 March 2010.

[47] Symetrics Industries. *Products: IDM-302*. n. pag. Web. 16 March 2010.

[48] Raytheon Company. *AN/ARC-164 UHF Airborne Radio*. n. pag. Web. 16 March 2010.

[49] R. A. Miller Industries, Inc. *The Antenna Professionals: UH-408 Antenna*. n. pag. Web. 16 March 2010.

[50] Lockheed Martin. *Theater Battle Management Core Systems: The "Engine" of the Air Operations Center*. n. pag. Web. 18 March 2010.

[51] GlobalSecurity.org. *Joint Tactical Radio System*. n. pag. Web. 18 March 2010.

[52] Compton, Matthew D., Kenneth M. Hopkinson, and Scott R. Graham. "The Network Tasking Order (NTO)," *MILCOM 2008:Unclassified Proceedings, San Diego, 17-19 November 2008*. CD-ROM. Piscataway NJ: IEEE Operation Center, 2008.

[53] Tiwari, Abhishek, Anurag Ganguli, Aditya Kothari, Sharath Avadhanam, Joseph Yadegar, Matthew D. Compton, and Kenneth M. Hopkinson. "Feasibility of Communication Planning in Airborne Networks Using Mission Information,"

MILCOM 2009: Unclassified Proceedings, Boston, 18-21 October 2009. CD-ROM. Piscataway NJ: IEEE Operation Center, 2009.

[54] Göçmen, Murat, Kenneth M. Hopkinson, and Matthew D. Compton. "The Benefits of a Network Tasking Order in Combat Search and Rescue Missions," *MILCOM 2009: Unclassified Proceedings, Boston, 18-21 October 2009*. CD-ROM. Piscataway NJ: IEEE Operation Center, 2009.

[55] Göçmen, Murat. *The Benefits of a Network Tasking Order in Combat Search and Rescue Missions*. MS thesis, AFIT/GCE/ENG/09-01. Graduate School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 2009 (ADA497812).

[56] Fair Isaac Corporation (FICO). *FICO Xpress Optimization Suite 7*. n. pag. Web. 24 March 2010.

[57] Ruckle, William H. *Modern Analysis: Measure Theory and Functional Analysis with Applications*. Boston: PWS-KENT Publishing Company, 1991.

[58] Menger, Karl. "Untersuchungen über allgemeine Metrik," *Mathematische Annalen*, 100: 75-163, December 1928.

[59] GAMS Development Corportation. *GAMS Home Page*. n. pag. Web. 25 March 2010.

[60] Cisco Systems, Inc. *DiffServ – The Scalable End-to-End QoS Model [Differentiated Services]*. Web. 19 March 2010.

[61] University of Southern California Information Sciences Institute. *The Network Simulator – ns-2*. n. pag. Web. 24 March 2010.

[62] Miras, Dimitrios. *A Survey of Network QoS Needs of Advanced Internet Applications (Working Document)*. Internet2 QoS Working Group, 2002.

[63] Falbe, Dan, Rodney Hayes, Richard McInnes, and Josh Morris. *Development of the Combat Search and Rescue (CSAR) Scenario to Support Modeling of the Joint Tactical Radio System's (JTRS's) Wideband Network Waveform (WNW)*, MITRE Technical Report 06B0000068. Hampton VA: The MITRE Corporation, August 2006.

[64] Defense Information Systems Agency. *JSC – SPECTRUM XXI*. n. pag. Web. 24 March 2010.

[65] Secretary of the Air Force. *Enterprise Network Operations Notification and Tracking*. Air Force Instruction (AFI) 33-138. Washington: SAF/XCIF, 28 November 2005.

Vita

Captain Matthew D. Compton graduated from Roncalli High School in Aberdeen, SD in 1992. He received the BA degree (*summa cum laude*) in mathematics and the MA degree in mathematics from the University of Oklahoma (OU) in 1996 and 1998, respectively. Prior to joining the Air Force, he was employed as an adjunct lecturer of mathematics at OU and Oklahoma City Community College. His first assignment after commissioning as an officer in 2004 was at the Office of Aerospace Studies at Kirtland Air Force Base, NM. He has been a PhD student of computer science since 2007 at the Air Force Institute of Technology, Wright-Patterson Air Force Base, OH. After graduating in 2010, he will be assigned to the Air Force Research Lab in Rome, NY. His main research interests are topology control, mobile ad hoc networks, and polymorphic networking.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
16-09-2010	Doctoral Dissertation	May 2007 – Sept 2010

4. TITLE AND SUBTITLE

Improving the Quality of Service and Security of Military Networks with a Network Tasking Order Process

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Compton, Matthew D., Capt, USAF

5d. PROJECT NUMBER
10-173

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/DCS/ENG/10-09

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory, Information Directorate, Rome Research Site
Attn: John D. Matyjas (john.matyjas@rl.af.mil)
525 Brooks Rd.
Rome, NY 13441
(315) 330-4255 (DSN: 587-4255)

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RIGF

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

This research presents a Network Tasking Order process that collects mission plans, network capabilities, and historical records to build a Network Tasking Order (NTO). The NTO document directs the form and usage of the network, much like an Air Tasking Order (ATO) directs the usage of air power. The NTO process is fleshed out with the content and format of the NTO given herein for the first time. Tools such as topology control algorithms are then shown through simulation to improve the quality of service of the network by finding favorable ways to connect the assets identified during the NTO process and to route the information through them, in one case preventing a 15% data loss. Furthermore, portions of the network can be hardened against cyber attack through a novel approach to polymorphic networking. The NTO process can provide a complete list of connections that are possible for a network. By periodically changing those connections in use and the routes taken through them, it becomes more difficult for adversaries to map the network in preparation for an attack. In the majority of cases, network availability to an attacker is reduced by more than 50%. It is also shown how existing topology control algorithms can be modified to produce heuristics for polymorphic networking.

15. SUBJECT TERMS

Global Information Grid, Network Security, Network Tasking Order, Polymorphic Networking, Quality of Service, Topology Control

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
REPORT	**ABSTRACT**	**THIS PAGE**			Dr. Kenneth M. Hopkinson (ENG)
U	**U**	**U**	**UU**	235	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636, x4579 kenneth.hopkinson@afit.edu

Standard Form 298 (Rev: 8-98)
Prescribed by ANSI Std. Z39-18